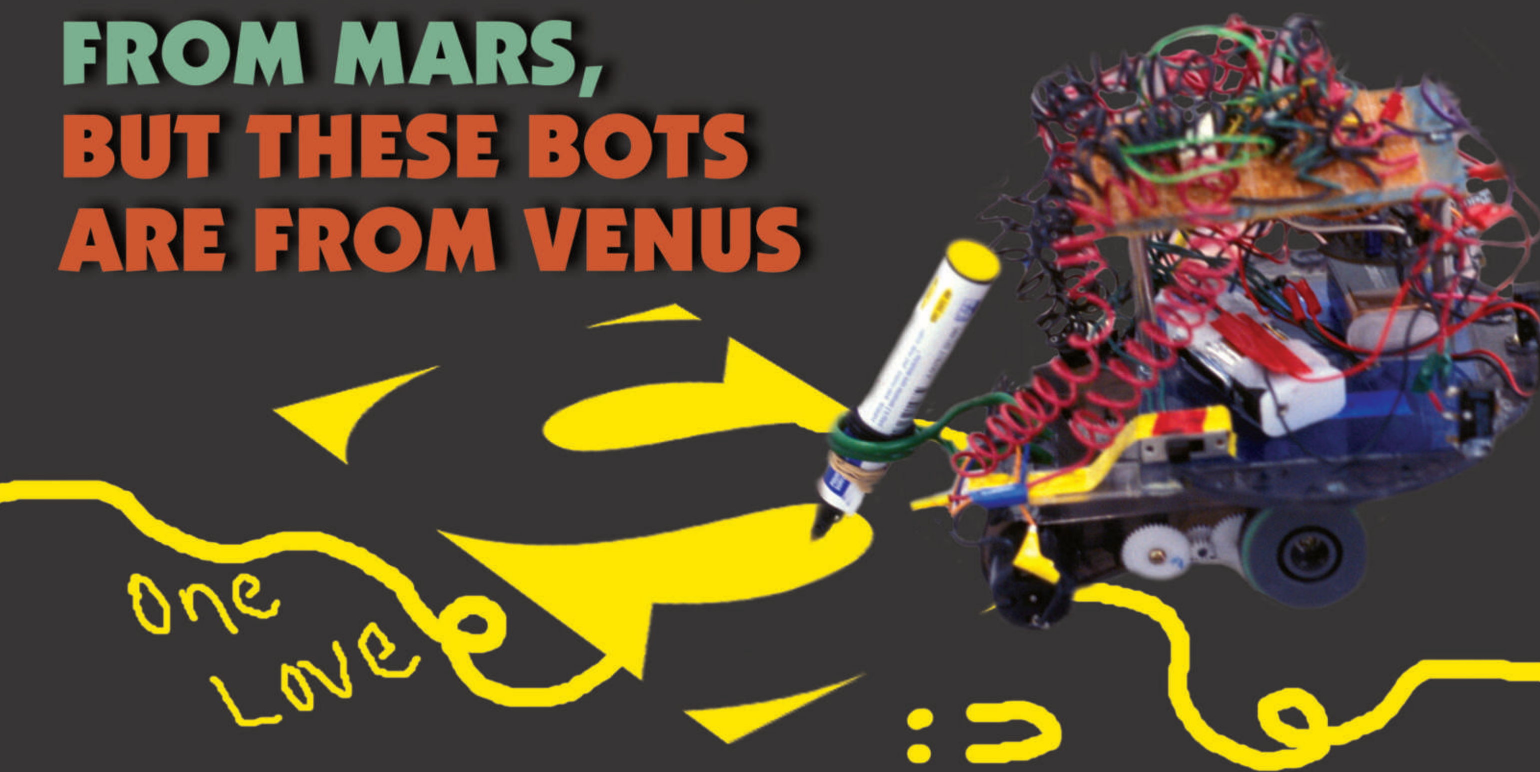
March 2004

SERVO MAGAZINE

Improve Life Through Science And Art

ROVERS ARE FROM MARS, BUT THESE BOTS ARE FROM VENUS



Engineering Hercules

Bot Booty: Is Less Really More?

The Taming Of The Screw

features

03.2004

SERVO Magazine (ISSN 1546-0592/CDN Pub Agree#40702530) is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. APPLICATION TO MAIL AT PERIODICALS POSTAGE RATE IS PENDING AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, 430 Princeland Court, Corona, CA 92879-1300** or Station A, P.O. Box 54, Windsor ON N9A 6J5. cpcreturns@servomagazine.com

table of contents

Vol. 2 No. 3

Cat meets Power Ranger. In Japan.

Coming 04.2004 in ***SERVO***

Mind / Iron

by Dan Danknick

As engineers, we often ignore the way much of the world feels toward technology. Encased in our solitary world of optimal decision making — the right batteries, proper metals, fast microcontrollers — it's a splash of ice water to encounter disinterest or its evil cousin, disdain.

When I was growing up, I viewed the hard sciences as the best tools that I could fill my mind with. I would ride to school and calculate how many Joules of kinetic energy were stored in my body/motorcycle system. Then, at a stop light, I would find the delta-T rise of the disc brakes. What a surprise to learn, as an adult, that 98% of the people I know don't care about that stuff, especially not an engineering unit with a silly name like "jewel". Most think of the pop singer.

I wonder if the personal robotics industry may be facing some of the same problems? As household robots try to gain ground, more and more people simply ask, "Why?" The burden of explanation falls back on our shoulders, since we're the futurists, as well as the implementers. This is a big deal and I fear we're ill-equipped to deal with this meta-technological question.

This month in *SERVO,* our authors approach this issue from different sides. Karl Williams proposes that certain physical features will make a robot accepted by humans. Ed Driscoll reports on Englebergers's current efforts in home healthcare. Roger Gilbertson tells of his recent trip to China and how the acceptance of robots begins in school, where they are introduced to students in a form as common as algebra. Finally, Bill Woolley writes in Appetizer about how his twin teenage sons have embraced robot building and its supporting technologies.

I still worry that "Why?" lingers. Perhaps it should. New technologies are always suspect during their rise toward utility. Railroads faced opposition from riverboat operators, who were less than keen to lose their shipping business; eventually, though, railroads became the spine of industry in the US by connecting isolated producers and consumers. So, by comparison, is iRobot's Roomba robo-vac struggling against Hoover and Electrolux? If Roomba doesn't "advance the art" in some useful way, then it is likely to fail.

What will make personal robotics useful? The answers are as varied as the applications. Ultimately, it will depend on the effort of innovators in our corner of science and their willingness to tackle the bigger problems. Every robotics club should have a standing challenge for its members, like a mini X-Prize: a non-trivial challenge that people can first dream about, then work toward solving. Competitions encourage invention, ideation, and creation.

That's the goal of our Tetsujin 2004 competition, disclosed on page 69. It's also the start of *SERVO Magazine*'s answer to "Why?" Augmented human strength is an ideal tool for rescue operations — prying open wrecked cars, rolling rubble out of the way, and stomping into a burning house to look for fire victims. A scaled down version can aid the infirm and injured.

Yes, I know this is an ambitious challenge. I'll do my part to present articles on the supporting technologies you'll need to compete in Tetsujin, but, in the end, you'll have to decide if you want take part in answering "Why?" and change how the world feels about robotics. If you think the exploration of Mars gets people excited, imagine what saving — or improving — a life will do.

If science is an amoral tool, then this is the grab handle for directing it toward something good. **SV**

Published Monthly By
The TechTrax Group — A Division Of
T & L Publications, Inc.
430 Princeland Court
Corona, CA 92879-1300
(909) 371-8497
FAX **(909) 371-3052**
www.servomagazine.com

Subscription Order ONLY Line
1-800-783-4624

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
robin@servomagazine.com

ADVERTISING SALES DIRECTOR
Rich Collins
rich@servomagazine.com

MANAGING/TECHNICAL EDITOR
Dan Danknick
dan@servomagazine.com

EDITORIAL ASSISTANT
Alexandra Lindstrom

CIRCULATION DIRECTOR
Mary Gamar
subscribe@servomagazine.com

WEB CONTENT/STORE
Michael Kaudze
michael@servomagazine.com

PRODUCTION/GRAPHICS
Rosa Gutierrez
Shannon Lemieux

DATA ENTRY
Karla Thompson
Debbie Lockett

OUR PET ROBOTS
Guido
Mifune

BIO-FEEDBACK

Dear SERVO:

Great magazine, folks. Keep up the great work. My two cents worth ... Robotics will take off in a big way when we can buy or build pre-engineered systems for motion control, vision, tactile sensory input, etc., so that each of us doesn't have to re-invent the wheel — so to speak — by spending countless hours duplicating work that's already been done.

Bill Black
via Internet

Dear SERVO:

This has probably been written about before, but I was fascinated years ago by the Odetics robot that walked like a spider. That was the robot that changed my attention from wheeled robots to legged ones. Perhaps *SERVO* could print an article reviewing the evolution of legged personal robots.

Gary Wilkerson
via Internet

Dear SERVO:

I want to thank *SERVO Magazine* for publishing the article dealing with robotic brains by Harold Reed. This article provided a new and interesting way to think about programming intelligence into robotic controllers. It is nice to see that *SERVO* provides an interesting array of information for its readers! I would also like to know if any other articles from Mr. Reed will be published in *SERVO*. Thanks for a great magazine.

Linda
via Internet

Dear SERVO:

Congratulations on this fine — almost perfect — magazine endeavor!

This is the first publication that has covered all the bases in only three issues. It's neither too elementary to put off the veteran or too complex to intimidate the tyro. Of special delight is the software/hardware amalgam and the robotics "art gallery."

As a suggestion for some future article, could you investigate the operation of the precursors to our electronic automatons? There were some very complex, cam-driven mechanical toys which are deserving of some analysis and expansion (such as ducks that lay eggs and monks that wrote sentences). There might even be some lessons to be learned for some of the complex movements we wish our humanoids to make, without adding servos.

Thanks for an incredible new view into robotics. Keep it coming...

P.S. About that "almost perfect" remark — can you make it a WEEKLY magazine?

John Dieudonne
via Internet

For Your Information ...

Although we foreshadowed a short article on the effort of Terra Engineering to finish their vehicle for the DARPA Grand Challenge, they were too busy to slow down for an interview. Additionally, at the time of publication, the vehicle had been broken down into a thousand pieces for final fit and function assembly.

We did manage to snap a picture of the IMU (inertial measurement unit) which Terra Engineering is using — reportedly a B-qual candidate from the advanced tactical fighter program. Worth more than double its weight in platinum, it sports a guaranteed 16 ms reset time after a nuclear "event."

That should be enough for an off-road race to Las Vegas! ***— Editor Dan***

Robots, Students, China, and the United States:

Cultivating The Robot Creators Of Tomorrow

by Roger G. Gilbertson

All around the world, robots inspire interest. People, especially children, respond with excitement, regardless of their economic or cultural background. You don't need a translator to comprehend a child's feelings upon meeting a robot face-to-face or triumphing in a robotics contest. This bodes well for all of us who are interested in robotics, as the inspired students of today will bring forth the amazing robots of the future.

I had never visited China. My first invitation to do so came recently from a company that produces the world's most widely used educational robotics system, the AS-M robot from Shanghai Grandar Electronics and Information Company. Since the year 2000, over 300,000 high school students have used the AS-M robot and as many as five million students have used the corresponding SVJC robot simulator software to learn first-hand about robotics. As in the US, Chinese schools have limited budgets, but even schools that cannot afford actual robots can successfully teach many aspects of robotics through the use of a robot simulation software package.

Grandar had hosted a robotics event in each of the three previous years, with each event drawing students from all over China. For November 2003, they planned the Fourth China Intelligent Robot Competition as their largest event yet. In total, it would draw nearly 600 students and over 100 educators, in addition to over 100 robots competing in a dozen games and events. I love to travel and my hosts were greatly encouraging, so I made arrangements to go.

Just Like New York?

Reactions to my upcoming trip varied greatly. “Robots in China? Huh. They have robots?” said one successful robot-building friend. At the other end of the spectrum, a friend and advisor who had just returned from a two week tour of China told me,”You'll be amazed! It’s just like New York!”

Now, this I could not comprehend — how could anything be at all like New York, let alone in one of the last standing communist countries and the nation with the world's largest population (1.3 billion compared to 0.3 billion in the US)?

I inventoried my impressions of China: rickshaws, rice fields, and Kodachrome-colored communist leaders standing on The Great Wall. My preconceptions of China seemed rooted in Nixon and Mao's meetings in the early 1970s. Then, just days before my scheduled departure, China sent aloft their first human to circle the Earth — Taikonaut Yang Leiwei, flying aboard the Shenzhou 5 spacecraft. Clearly, my preconceptions of China needed updating.

Sitting in the window seat of the 747-400, my own flight seemed to brush the edge of space and I gazed down through a purple twilit haze into the impossibly frozen and remote Arctic regions of Alaska and Kamchatka; I knew my views would soon change. After 10 hours, we descended into warmth and light — and into a world I could not have imagined.

Shanghai China, 2003

To imagine Shanghai, take a map of New York City, photocopy it, then cut-and-paste two or three of them together, packing them along either side of a great river. Sprinkle with dozens of ultra-modern skyscrapers and a wide variety of bold new civic centers, museums, and sporting complexes straight out of *The Jetsons*. Next, you need to lace it all with ribbons of surprisingly western streets, parkways, freeways, and skyways (illuminated at night from underneath with blue lighting effects). Then, post large neon billboards wherever space permits. Next, pack the whole place with all manner of pedestrians, wooden carts, bicycles, gas and electric scooters, cars, busses, and trucks from all eras, smoothly interweaving with each other in a fluid, modern dance of traffic and transit.

The ultra-modern Shanghai Pudong Airport Terminal Building rivals similar structures in any country and, if you arrive during its operating hours, a high-speed magnetic levitation monorail system will whisk you from the airport to the city. Shanghai has the first and only commercial system of this kind in the world. What a world — I wondered why we in the west simply don't know about this?

• • Kids are Kids — Professor Zhyi and students from ShangDe Experimental School, standing with the author.

The Events

My host, Dr. Yun Wei-Min, founded Shanghai Grandar soon after graduating from university. He quickly became recognized as a leading innovator and entrepreneur in China. In 1998, the Chinese government selected Dr. Yun to meet and lunch with then US President Bill Clinton.

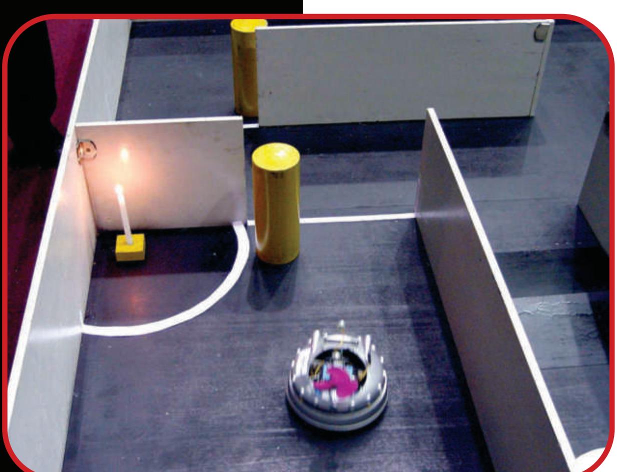
上海

Dr. Yun had observed that robotics combined so many areas of technology and that students naturally responded to robotics with great energy. They surveyed all the available products for robotics education, then set themselves to pursue the goal of developing a better system. They created the AS series of robots and convinced the government of

• • The Gymnasium

• • RoboCup Jr. robot soccer. Two on two — push the glowing ball into the opponent's goal to score points.

• • The Connecticut Robot Fire Fighting event. Find the candle burning somewhere in the maze, extinguish it, then return to the start in the shortest period of time.



China to try it as an educational "demonstration" program. The rest is history, as the tests proved highly successful; robotics have since been integrated into China's national high school curriculum. In some provinces of China, every high school student studies robotics for at least one semester.

The morning of the Fourth China Intelligent Robot Competition, I ventured across town from my thoroughly western hotel room to the Shanghai East Normal University, the site of the games. The university's large gymnasium already seethed with students — from 10-year-olds to college students. Hundreds of pupils and robots, accompanied by educators, converged from nearly every province of China — including Hong Kong, Macau, and even Taiwan. In addition, key educators from the Chinese government attended and spoke to the crowd on a variety of topics.

The dozen robot events included many popular international robot games — most notably, RoboCup Jr. style soccer and the Connecticut Robot Fire Fighting challenge. In addition, many students competed in a high-pressure "Day of the Event" competition. Here, the judges revealed a new robotic challenge on the morning of the event, then students worked fervently for most of the day to assemble, program, and test their systems. Next, they had just one chance to successfully demonstrate their solutions before the judges. One did not need a translator to understand the expressions of elation or despair on the contestants' faces as their robots either succeeded or failed at the assigned task.

In all, the Intelligent Robot Competition gathered a great amount of media coverage. Eleven articles appeared in seven different newspapers and seven separate news reports appeared on four different national and local television channels. Grandar plans even more events with larger attendance potential, including the next competition, which is scheduled for November of 2004.

Made in China

Perhaps not surprisingly, nearly every one of the robots in competition was based on the Grandar AS-M robot and related models. However, as with any quality robot system, it lent itself to the modifications and additions which many students performed on their machines, often to the point of complete transformation!

Other than its wide availability in China and its status as the standard educational platform, what makes the AS-M robot so readily adaptable to such a wide range of robotic challenges?

The AS-M robot arrives fully assembled, tested, and ready to go, letting students (and teachers) gain nearly immediate success at getting a robot up and responding. Without needing to assemble a kit of parts, solder, assemble circuits, or do extensive programming, students gain initial successes that increase their confidence and keep them motivated for more challenges.

The AS-M robot operates on a Motorola 68HC11 processor and the innovative programming language teaches object-oriented programming, as well as C language. The AS-M hardware suite includes:

- Eight-zone bump sensor skirt
- Reflected IR object detection
- Photo sensors
- Speaker (not a buzzer)
- Microphone
- Dual drive motors
- Dual optical wheel encoders
- Third motor driver on board (for your own motor)
- Three-wire R/C-type servo motor port
- Full hardware bus access to HC11 processor

In addition, Grandar also produces many available expansion boards and accessories.

On the main electronics board, all integrated circuits are socketed and replaceable by the user! This means no dead robots in

the classroom or home lab, as faults can be diagnosed and repaired with basic electronics tools and procedures. The robot's documentation includes complete schematics for all of its systems. The AS-M package even includes a plug-in proto board and connectors, enabling more advanced students and experimenters to add hardware projects of their own design.

The AS-M system meets the needs of a wide range of roboticists. Beginners learn the robotic basics of electronics, mechanics, and software programming. The AS-M's VJC graphical programming language teaches logic and proper design right from the start. For intermediate students, the AS-M's unique interchangeable sensor system lets users create sophisticated solutions with minimal delay. More experienced programmers can actually "lift the hood" on the graphical programming language and work directly with the generated C code, giving them complete control. The VJC software runs with versions of Windows (98 onward) and good results have even been acheived using it on a Macintosh with Virtual PC software and a USB-to-serial converter.

For advanced roboticists, the AS-M's expandable hardware permits the addition of all kinds of circuitry, sensors, output devices, and more. Advanced programmers can work directly in C for maximum power and flexibility.

What About the US?

The AS-M robot's thoughtful design and highly practical implementation make it ideal for students from a wide range of ages and experience levels. I found the entire package so impressive that we've arranged to bring it to the US, via our **RobotStore.com** website and catalog. The software and documentation have been adapted to English and several independent groups in the US have created educational curriculum packages for schools and even home school students.

In these days of restricted school budgets and rising demands for improved technology education, the AS-M robot provides students, teachers, principals, and parents with the opportunity to reach new levels of academic achievement. Why? Because robots are cool! Robotics motivates students and motivated students learn better, enjoy what they learn more, and achieve higher academic results in all areas.

In China, students that succeed in science and technology activities (such as robotic competitions) receive additional academic credit, greatly increasing their access to the best of higher education. The government plans to reform the existing technology education system from knowledge-teaching to ability-cultivating. Rather than just learning how to make good PowerPoint presentations (Yes, Bill Gates' reach even extends into Mainland China.), students will learn how to develop, create, and evaluate their own software and other technological creations.

In both China and the US, parents believe that information technology courses are of great value to students. Schools that provide more exciting and fulfilling programs produce more capable students, who, in turn, have a wider range of career choices and options for the future.

Robotics rests at the convergence point of nearly all our modern technologies and provides an exciting vehicle that supports learning in all areas of science and technology, as well as many areas of the humanities, in addition to teaching teamwork building and even international co operation.

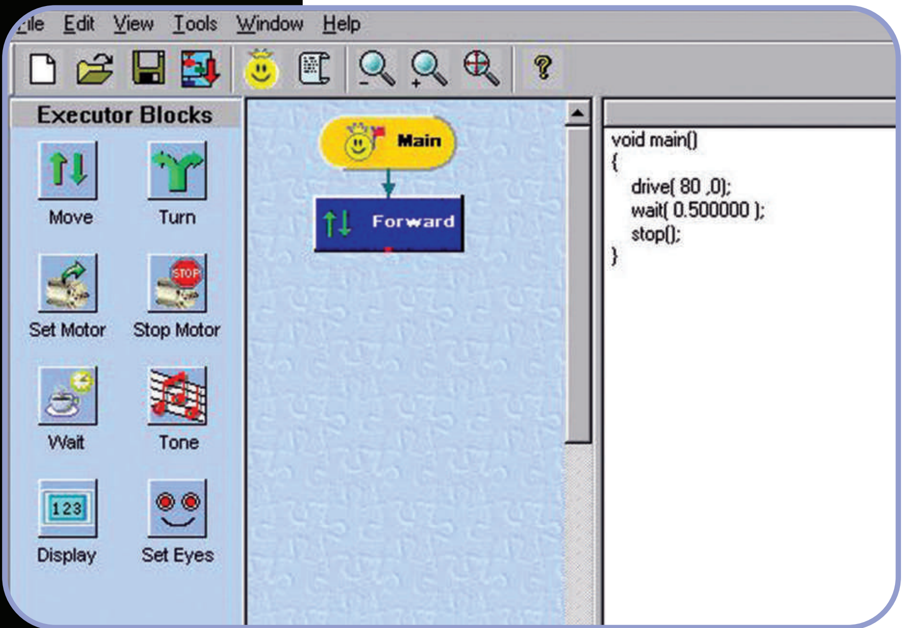
Imagine

As a representative of the US robot-building community and the only foreigner in attendance, my hosts asked me to join in the awards ceremony and give a brief speech during the closing.

I spoke of how the moon landings of the 1960s greatly inspired to me to study science, engi-

• • "Day of the Event" challenge. Build, program, and operate an autonomous robotic delivery vehicle. The challenge was announced on the morning of the event and each team had just one chance to perform before the judges!

• • Grandar AS-M Robot



• • The VJC programming environment for the AS-M robot. Users can work in graphical "flow chart" mode or "lift the hood" and work in the resulting C code (above, right). It is powerful, flexible, and expandable to fit the growing skill of the experimenter.

• • The Presidium — Leading robotics and technology educators from all over China (and one robot builder from the US) present awards to the day's many winners.

neering, and, eventually, robotics. I compared my feelings to the excitement and inspiration that they must feel at the successful mission of Yang Leiwei and the Shenzhou 5.

I challenged them to strive for great achievements of their own, to reach for the moon, Mars, and beyond, but to always keep in mind the warning of Marshall McLuhan — the great Canadian philosopher of technology — who said, "Every technological extension has a corresponding amputation." Just as a step ladder can help a person reach higher, it also makes falling more dangerous, so we must always choose carefully what we want to build.

We must all, no matter what our place in the world may be, strive to create more and greater opportunities to enable students of all abilities and backgrounds to experience the full range and power of our technologies. After all, if our children cannot fully command the technologies of the future, who will?

The people of China see the amazing benefits that technology has brought the western countries and they, of course, want the same and better for themselves and their children.

Just like us, they want robots to take over the dangerous, boring, and repetitious jobs.

So, as robot builders and the teachers of tomorrow's robot builders, let's keep in mind how our efforts today will affect the world we will live in tomorrow. Every embodiment of technology, from a sharpened stick to a space station, has the potential to make our lives better and robotics provides perhaps the greatest promise of all technologies.

I find envisioning a world where robots give everyone the opportunity to pursue his or her highest potential to be far more challenging than picturing what I was to encounter on my first trip to China.

Clearly, working to create such a bright robotic future will be worth the effort and we should even have some fun along the way.

Build more robots! **SV**

About the Author

In college, Roger G. Gilbertson studied engineering, robotics, and the walking patterns of living creatures. In 1987, he co-founded Mondo-tronics, Inc., to explore the commercial applications of Shape Memory Alloy wires; in 1995, he launched RobotStore.com*, the internet's first commercial robotics site. Mondo-tronics' Robot Store continues to lead the field in presenting the best and most innovative new robot products for students, educators, hobbyists, and experimenters. Roger lives and works in Marin County, CA, where robots still prefer to wait for humans to provide instruction on what to do, rather than take their own initiative.*

Programming A Mini-Sumo Robot In Java

by Steve Grau

Software development technology has progressed significantly over the past 40 years. In just the last 10 years, object-oriented programming has gained widespread acceptance for commercial applications.

Still, most hobbyists find themselves using older procedural programming methods. This article explains how Java, a modern object-oriented language, was used to program a mini-sumo robot.

Theory of Operation

To be effective, a mini-sumo robot must be able to do the following:

- Stay on the dohyo (competition ring).
- Hunt for the opponent.
- Target (aim at) the opponent.
- Attack the opponent.

Here is how the mini-sumo robot accomplishes these things:

Sensing the Dohyo Edge

The mini-sumo robot relies on two Fairchild QRB1134 infrared photo-reflectors, mounted at the left and right front corners of the robot to detect the white edge of the dohyo.

Sensing the Opponent

Two forward-facing Sharp GP2D12 infrared range sensors, mounted on the front of the robot, are used to sense and target the opponent.

Moving and Steering the Robot

The robot's motion is controlled using two servo motors, which are modified for continuous rotation.

Steering is accomplished by applying more power to one wheel than the other, causing the robot to arc left or right, or by powering the wheels in opposite directions, causing the robot to rotate in place. The pulse width on the servo's control input can vary the power applied to the wheels (Figure 1).

Controlling the Robot's Behavior

Software implements the robot's intelligence, linking its behavior to the data it collects from its surroundings via its four sensors. The mini-sumo program discussed in this article implements a finite state machine (Figure 2) to control the robot's behavior. As shown in Table 1, there are four states: SURVIVE, HUNT, TARGET, and ATTACK. The arrows in Figure 2 depict events that cause transitions from one state to another. A detailed description of each state is provided in Table 1.

State	Description	Actions
SURVIVE	The robot enters this state when it detects the dohyo edge. Its goal is to survive by not going off the dohyo. The robot rotates away from the sensor that sensed the edge to face back toward the center of the dohyo.	1. Rotate away from the line sensor which sensed the edge. 2. Switch to HUNT state when rotation is complete.
HUNT	The robot is not at the edge of the dohyo, but hasn't sensed the opponent. The robot moves around in an arcing pattern, so that its range sensors will sweep across the ring in hopes of sensing the opponent.	1. Switch to SURVIVE state if the dohyo edge is detected. 2. Switch to TARGET state if range sensors indicate an object ahead. 3. Otherwise, drive in an arc by applying more power to one wheel than the other.
TARGET	The opponent has been sensed ahead. Aim the robot to face the opponent directly.	1. Switch to SURVIVE state if the dohyo edge is detected. 2. If still sensing opponent, but opponent is not directly ahead, turn slightly to aim at opponent. 3. If opponent is directly ahead and close, switch to ATTACK state. 4. If opponent is ahead, but not close, move straight forward. 5. Otherwise, switch to HUNT state.
ATTACK	The opponent has been found and aiming is complete. Drive straight ahead at full power to push the opponent off the dohyo.	1. Switch to SURVIVE state if the doyho edge is detected. 2. Otherwise, drive straight forward at full power.

TABLE 1 — State Machine Summary

Mini-Sumo Robot Hardware

The mini-sumo robot was constructed using the Sumo11 controller and expansion board, the Mark III chassis, and the sensors previously described. Table 2 provides a listing of the hardware components used.

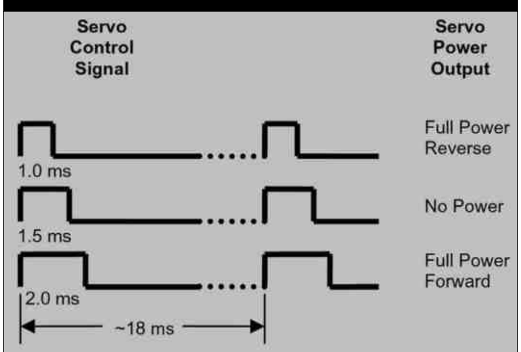
Sensor and Servo Wiring

The sensors and servos were connected to the Sumo11 controller (Figure 3). The servo motors were connected by plugging the preinstalled connector into the servo header with the servo control signal (white wire) attached to the pin marked "S". The right servo is attached to servo port 3 and the left servo to port 0.

The right and left line sensors were soldered directly to the Sumo11 main board to the ports marked "R-LINE" and "L-LINE", respectively. The wire order, from left to right, is white, blue, green, and orange.

The left and right range sensors were connected to analog inputs 3 and 2, respectively, using the JST cables with a custom plug attached opposite the JST connector, as shown in Figure 4.

FIGURE 1



Servo Control Signaling

FIGURE 2

Mini-Sumo Software State Machine

Mini-Sumo Robot Software

The mini-sumo software was created using the RoboJDE software development environment. This development environment was installed on a host PC with a serial port connecting it to the Sumo11 controller. The RoboJDE virtual machine was then downloaded to the controller card to enable the controller to run Java programs. Once these steps were completed, RoboJDE was used to edit, build, download, and debug the mini-sumo robot control program. Complete source codes for the mini-sumo program can be downloaded from **www.ridgesoft.com**

The Java programming language uses a syntax derived from the C language. However, Java includes object-oriented programming features, which are not found in procedural programming languages, such as C and BASIC (see the Java Programming Concepts sidebar for a summary). For more in-depth information on programming robots using Java, visit the RidgeSoft website, listed previously.

Software Implementation

The `BasicMiniSumo` class (see source file **BasicMiniSumo.java**) implements all of the control algorithms in a way that is independent of the actual robot controller board being used. A second, trivial class, named `BasicMiniSumo11`, implements the specific controller board details, such as port and sensor associations. This arrangement makes the majority of the program independent of the controller hardware, minimizing the effort to port the program into other controller boards in the future.

`BasicMiniSumo11.main()` Method

The application starts in the `BasicMiniSumo11.main()` method by retrieving a reference to the `Display` object from the class `HandyBoard`, using the following statement:

```
Display display = HandyBoard.getLcdDisplay();
```

Note: The Sumo11 is `HandyBoard` compatible. Features of the Sumo11 controller are accessed via the `HandyBoard` class.

The `Display` object provides a method to print text on specific lines of the LCD screen. The program identifies itself to the user with the following statement:

```
display.print(0, "BasicMiniSumo11");
```

Following this, an object of the `BasicMiniSumo` class is instantiated. (Instantiating a class in Java creates an object — which is an instance of the class — by allocating memory for the object and initializing the object.) Using the `new` operator to allocate the memory and call the constructor method (the method named `BasicMiniSumo`) accomplishes this. References to a number of objects are passed to the constructor method, in addition to a number of constants that may be used to tune the robot's behavior. The `BasicMiniSumo` class requires `Motor` objects for the two drive motors. However, the robot actually uses servos that have been modified for continuous rotation. The `ContinuousRotationServo` class implements the `Motor` interface to an encapsulated `Servo` object. This creates a façade that makes a `Servo` object controllable as a `Motor`, allowing motor control software to control the modified servo, just as it would control a conventional gear motor. The `ContinuousRotationServo` class also allows the sense of rotation to be reversed, which is necessary for the right servo. Forward rotation for the right servo is actually reversed from the robot's perspective. The left servo is attached to servo port 0 and the right servo is attached to servo port 3. The `Motor`

Qty	Part	Supplier
	Controller	
1	Sumo11 main board	**www.1sorc.com**
1	Sumo11 expansion board	**www.1sorc.com**
	Chassis, Sensors, and Actuators	
1	Mark III chassis kit	**www.junun.org**
2	Servo (GWS SO3N 2BB)	**www.junun.org**
1	Pair of molded wheels	**www.junun.org**
2	Sharp GP2D12 range sensor	**www.junun.org**
2	Fairchild QRB1134 IR sensor	**www.junun.org**
2	JST cable for GP2D12	**www.junun.org**

TABLE 2 — Mini-Sumo Hardware

Java Programming Concepts

Abstraction
Hiding details of an object that are not relevant to software interfacing to the object minimizes interdependencies, making it easier to implement and maintain software.

Encapsulation
Data and functions of a particular type of object are bundled into a single package, called a "class", that has a well-defined interface. The interface restricts access, such that software interfacing to an object can only do what the interface will allow, preventing objects from being operated on in ill-defined ways.

Class
A class encapsulates the implementation for one type of object. A class defines the member variables and implements the methods for a particular type of object.

Object
An object is an instance of a class. The mini-sumo program uses two objects to interface to the left and right servo motors. The `Servo` class implements these objects.

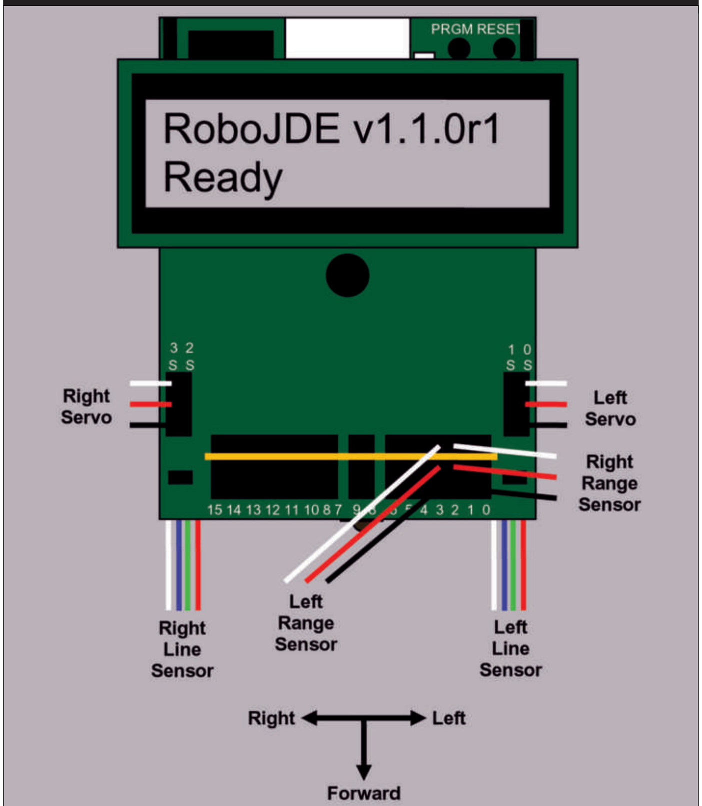
Member Variable
Member variables are variables unique to a specific object.

Method
Methods are the object-oriented equivalent to functions, procedures, and subroutines in other languages. However, methods are always implemented as part of a class.

Interface
An interface is a list of methods that, when implemented by a class, provides a subset of functionality that is useful, independent of other methods of the class.

Reference
A reference is a type-safe pointer to an object.

FIGURE 3



Sensor and Servo Connections to Sumo11 Controller

façade objects are created by the following statements:

```
new ContinuousRotationServo(HandyBoard.getServo(0), false)
new ContinuousRotationServo(HandyBoard.getServo(3), true)
```

The `BasicMiniSumo` class also requires `AnalogInput` objects for each of the sensors, which are obtained as follows:

```
HandyBoard.getAnalogInput(6)   // left line sensor
HandyBoard.getAnalogInput(5)   // right line sensor
HandyBoard.getAnalogInput(3)   // left range sensor
HandyBoard.getAnalogInput(2)   // right range sensor
```

References to these objects are passed to the `BasicMiniSumo` constructor.

The constructor initializes the `BasicMiniSumo` object's member variables, `mLeftMotor`, `mRightMotor`, `mLeftLine`, `mRightLine`, `mLeftRange`, and `mRightRange`, respectively, with these references.

Once the `BasicMiniSumo` object is constructed and initialized, control is passed to it by calling its `go()` method:

```
sumo.go();
```

`BasicMiniSumo.go()` Method

This method calls the `displaySensorReadings()` method to start the sensor test. This test allows the raw sensor reading from each of the sensors to be viewed by scrolling between the four sensors using the thumbwheel on the Sumo11 controller. The sensor test terminates when the start button is pressed. When the start button is released, the five-second-countdown loop starts. (At the start of a mini-sumo match, the robot is required to sit idle for five seconds after the contestant steps away from it.) The program loops five iterations, displaying the number of seconds remaining, and sleeping 1,000 milliseconds (one second) each time through the loop. The sleep statement is as follows:

```
Thread.sleep(1000);
```

Finite State Machine Control Loop

When the countdown is complete, the program enters an infinite while loop that executes the control state machine until the robot is turned off or reset, as shown in the following code fragment:

```
while(true) {
      …                // sensor sampling code
    switch (state) {

    case SURVIVE:
        …       // survive code
        break;

    case HUNT:
        …       // hunt code
        break;

    case TARGET:
        …       // target code
        break;

    case ATTACK:
        …       // attack code
        break;
    }
}
```

Each time through the loop, the program samples the sensors and then switches to the code for the current state.

Sensor Sampling

The line sensor analog input is sampled and converted to a Boolean value, which is `true` if the sensor is detecting the edge of the dohyo. Sampling the `AnalogInput` and comparing it to a threshold value accomplishes this. If the reading is below the threshold value, it is assumed that the sensor is over the edge of the dohyo. (The sensor reading is lower when over white than when over black.) This is done as follows:

```
boolean atLeftEdge = (mLeftLine.sample() < mEdgeThreshold);
```

```
boolean atRightEdge = (mRightLine.sample() < mEdgeThreshold);
```

The range sensors are analog inputs. The sensors will read high when close to an object. The range sensors are sampled, then the average and the difference of the two readings are calculated using these statements:

```
int leftRange = mLeftRange.sample();
int rightRange = mRightRange.sample();
int rangeDifference = leftRange - rightRange;
int rangeAverage = (leftRange + rightRange) / 2;
```

If the opponent is sensed, the `rangeAverage` will be higher than when the opponent is not in front of the sensors. The `rangeDifference` is used to target the opponent. If one sensor senses the opponent, but the other does not, the `rangeDifference` will be high. If the opponent is straight ahead, the `rangeAverage` will be high and the `rangeDifference` will be low. Each state needs to check for the edge of the dohyo. To avoid duplicating the check in several states and to accelerate transitions to the SURVIVE state, this check is done before the switch statement:

```
if (atLeftEdge || atRightEdge)
      state = SURVIVE;
```

The behavior of the robot in various states is described in the Theory of Operation section. The following sections provide the code fragments for each state.

SURVIVE State

```
    if (atLeftEdge) {
          rotate(-135);
          searchClockwise = true;
    }
    else if (atRightEdge) {
          rotate(135);
          searchClockwise = false;
    }
    state = HUNT;
```

HUNT State

```
    if (rangeAverage > mTargetThreshold)
          state = TARGET;
    else if (searchClockwise)
          arc(mHuntPower, mHuntFactor);
    else
          arc(mHuntPower, -mHuntFactor);
```

TARGET State

```
    if (rangeDifference > mTargetThreshold)
          spin(false, mSpinPower);
    else if (-rangeDifference > mTargetThreshold)
          spin(true, mSpinPower);
    else if (rangeAverage > mAttackThreshold)
          state = ATTACK;
    else if (rangeAverage > mTargetThreshold)
          forward(mTargetPower);
    else
          state = HUNT;
```

ATTACK State

```
    forward(mAttackPower);
```

Navigation Methods

The `BasicMiniSumo` class contains the following methods, which are used to navigate by manipulating the motor power:

- `forward(int power)`
- `arc(int power, int factor)`
- `spin(boolean clockwise, int power)`
- `rotate(int degrees)`
- `stop()`

Forward

The `forward` method attempts to move the robot straight forward by setting the power to both motors to the same level.

```
mLeftMotor.setPower(power);
mRightMotor.setPower(power);
```

Arc

The `arc` method attempts to move the robot forward, arcing to the left or right by applying slightly more power to one motor than the other.

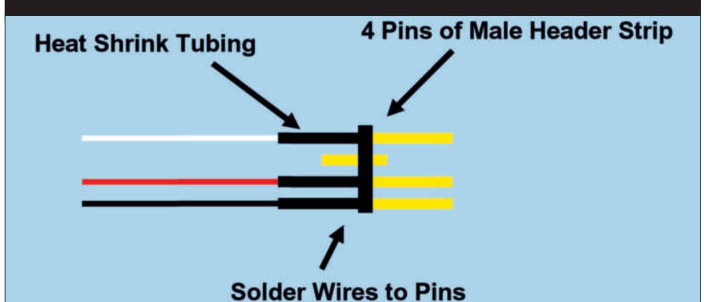
```
mLeftMotor.setPower(power + factor);
mRightMotor.setPower(power - factor);
```

Spin

The `spin` method attempts to spin the robot in place, without a specified stopping point, by running the motors in opposite directions.

```
if (clockwise) {
        mLeftMotor.setPower(power);
        mRightMotor.setPower(-power);
}
```

FIGURE 4



Range Sensor Connector Plug Construction

```
else {
        mLeftMotor.setPower(-power);
        mRightMotor.setPower(power);
}
```

Rotate

The `rotate` method attempts to rotate the robot a specified number of degrees by running the motors in opposite directions for a fixed time per degree of robot rotation.

```
if (degrees < 0) {
        degrees = -degrees;
        mLeftMotor.setPower(mRotatePower);
        mRightMotor.setPower(-mRotatePower);
}
else {
        mLeftMotor.setPower(-mRotatePower);
        mRightMotor.setPower(mRotatePower);
}
try {
        Thread.sleep(degrees * mRotateFactor);
}
catch (InterruptedException e) {}

stop();
```

Stop

Turns off the motors.

```
mLeftMotor.setPower(0);
mRightMotor.setPower(0);
```

Enhancing the Mini-Sumo Robot

The program presented here implements a fairly simple control algorithm. There are many alternatives to experiment with to improve the robot's competitiveness. The following list includes some ideas for potential improvements:

- Tune the control constants to enhance the robot's performance.
- Improve aiming in the targeting state.
- Add steering in the attack state.
- Add more sensors to enable detection of the opponent from any direction.
- Add sensors to detect being pushed backward off the dohyo.
- Add shaft encoders to sense and better control the robot's motion.
- Add different types of sensors, such as sonar.
- Add new states, such as escaping when being pushed by the opponent.
- Implement a behavior system using the `Behavior Arbiter` class.
- Experiment with new motion algorithms in the existing states.
- Randomize the behavior of the robot to make the robot less susceptible to weaknesses of a particular behavior.
- Use multi-threading to execute portions of the control software concurrently, for example, sampling sensors at the same time as executing maneuvers.

Conclusion

This simple state machine control algorithm was easy to implement in Java. The program proved to be effective at keeping the robot on the dohyo, as well as finding, targeting, and attacking the opponent. Programming in Java allowed pre-built robotics software components from the RoboJDE class library to be used to control servo motors and interface to sensors. The strong compile-time and run-time error checking of Java resulted in no difficult debugging problems or controller crashes, which are often common when using other programming languages. **SV**

ASK MR. ROBOTO

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are. And what more would you expect from a complex service droid?

roboto@servomagazine.com
Our resident expert on all things robotic is merely an Email away.

Q. Do you know of any good places on the Internet where I could find information about the different types of robots that people are building?

— Charles Bahr
via Internet

A. Personally, my favorite place to visit to see what everyone is creating is the Robot Menu website at Arrick Robotics (**www.arrick.com/robomenu**). There, you will see just about every type of robot imaginable.

People from around the world submit their robotic creations to be displayed. The site includes a nice photo of each robot, along with a short description about what the robot does, design challenges, and costs. Often, there is a link to the creator's website for additional information. At last count, there were over 420 different robots on this site! You can spend days digesting all of the information on this website. Many thanks to Roger Arrick for hosting this wonderful resource that all robot builders can learn from.

Q. Mr. Roboto, I recently bought an SRF04 ultrasonic sensor and I would like to know how to calculate the distance conversion factor for a BASIC Stamp 2p.

The example program that came with my SRF04 was for a BASIC Stamp 2. Through trial and error, I was able to figure out one that works, but I would like to know if there is a better way to do this.

— Will Harrison
Los Angeles, CA

A. The Devantech SRF04 (**www.robot-electronics.co.uk**) ultrasonic range finder uses a short 40 kHz sound wave to measure the distance of obstacles from 3 cm to 3 m (1.2 in to 9.8 ft). The object's distance is calculated by measuring the time it takes for a sound wave to travel from the sensor to the object, multiplied by the speed of sound. The speed of sound in dry air at room temperature (20° C / 68° F) is 343.4 m/s (768.2 mph).

The SRF04 outputs the total round-trip time of the sound wave to the object and back.

If the relative speed of the object, with respect to the sensor, is small (i.e. less than 1 or 2 percent of the speed of sound, which is less than 5 m/s), we can safely assume that the amount of time it takes for the sound wave to reach the object will be half the total SRF04 output time. Equation 1 shows a simple equation for calculating the distance to the object where d_{obj} is the distance to the object, V_s is speed of sound, and T is the total pulse width time from the SRF04.

$$d_{obj} = V_s\left(\frac{T}{2}\right)$$

Equation 1

Most microcontrollers measure pulse widths as some integer number of integral clock time periods, which we will call N for the number of time periods. The unit of time for each one of these time periods, P, is unique for each microcontroller. Thus, the total measured pulse width time, T, is the number of time periods, N, multiplied by the unit of time for each time period, P (T=N*P). By plugging this into Equation 1, we have an equation that

$$d_{obj} = \left(\frac{V_s P}{2}\right) N$$

Equation 2

works better with microcontrollers.

Since the speed of sound, V_s, and the unit of time for each time period, P, are constants, these terms can be grouped into a term called the "conversion factor". Equation 3 shows two simplified versions of the distance formula as functions of the measured number of time periods, N, and a conversion factor. The conversion factor, K, is V_s multiplied by P and divided by 2 ($K=V_s*P/2$). The conversion factor C is just 1/K, and 1/C is equal to K.

$$d_{obj} = K*N \quad \text{or} \quad d_{obj} = N/C$$

Equation 3

If your microcontroller can handle floating point numbers, then use the K conversion factor relationship. If your microcontroller can only handle integer numbers, then use the C conversion factor relationship.

Table 1 shows several conversion factors for various microcontrollers that will output distance measurements in inches and centimeters.

Both floating point and integer

conversion factors are shown in this table, though not all microcontrollers can use both numbers.

Hopefully, this short discussion has shown you an easy way to calculate the distance to an object using the SRF04 ultrasonic range finder. This analysis can also be applied to all ultrasonic range finding sensors.

Table 1. SRF04 distance conversion factors for various microcontrollers.

		Inches		Centimeters	
Microcontroller	P	K	C	K	C
BASIC Stamp 1	10 µs	0.06760	15	0.17170	6
BASIC Stamp 2	2 µs	0.01352	74	0.03434	29
BASIC Stamp 2sx	0.8 µs	0.00541	185	0.01374	73
BASIC Stamp 2p	0.75 µs	0.00507	197	0.01288	78
Javelin Stamp	8.68 µs	0.05868	17	0.14904	7
Basic Atom	1 µs	0.00676	148	0.01717	58
OOPic	0.2 µs	0.00135	740	0.00343	291
Brainstem	1.6 µs	0.01816	92	0.02747	36
BasicX	1.08 µs	0.00734	136	0.01863	54

Q. To control the motors of my robot, I have been looking at R/C car electronic speed controllers — the ones you plug into a servo port of an R/C receiver that can control higher amp motors. One is the Vantec RDFR47E on the bottom of Page 12 in the January, 2004 issue of *SERVO*. What I do not understand is the ampere rating of these units.

These controllers look like they have #14 or #12 gage wires attached to the units via spade connectors. These units are rated at 50 to 60 amps. How do wires of this size carry this amount of current without overheating? Thanks.

— Joe Fishback via Internet

A. The Vantec (**www.vantec.com**) motor controllers are some of the best motor controllers available on the market today. When they advertise a particular continuous current rating, they mean continuous. The typical R/C electronic speed controllers that you can purchase at your local hobby store advertise maximum current ratings that are more theoretical under ideal conditions and are not to be exceeded for more than a second or two. Otherwise — poof — the magic smoke escapes.

The wires that are shown in the advertisement are for connecting signal control lines to standard R/C radio receivers or microcontrollers. The RDFR47E has an impressive maximum continuous current rating of 75 amps! Vantec recommends using a #8 gage wire when operating with this much current. The barrier terminal strip on this controller uses 6-32 screws.

The crew from Vantec told me that it can be challenging to get enough copper from the #8 gage wires connected to the controller. They told me that some of their customers have successfully used a pair of over-sized crimp-style ring connectors, which are connected back to back and squeezed down to form an oval so that it can fit between the barriers.

Vantec also recommend that, if crimp style connectors are used, they should be soldered to the wires. The #8 gage wires are recommended to minimize the voltage drop under high current loads in the wires between the motors and motor controller.

Smaller diameter wires can be used if the wiring runs are shorter. Keep in mind, however, that drawing extremely high currents through any wire will result in the wires getting very hot. Silicone- or Teflon-coated wires are highly recommended for high current wiring.

Q. My high school team is planning to build a robot with a unique drive system that requires a differential (like the one found on cars), but I'm having a lot of trouble finding one that we can use.

We are planning to connect the differential either directly to the wheels or through a small gear reduction. The robot will weigh about 100 pounds and must be able to withstand impacts with other robots. Basically, the differential needs to be light, durable, and fairly efficient.

I've searched at least 20 of the websites listed in my *Robot Builder's Sourcebook*, but only differentials with 1/4 inch — or even smaller — shafts are available. I also looked at the NPC Robotics website, but I couldn't find any differentials for drives. I would like something with at least 1/2 inch shafts, since this worked fine previously with our traditional tank-style drive system.

We would greatly appreciate it if you could give us the names of some suppliers that sell differentials that will meet our needs. At this point, almost any differential with 1/2 inch shafts will do. Differentials with a drive shaft gear or even a drive sprocket can be incorporated into the design. Thanks.

— Paul
via Internet

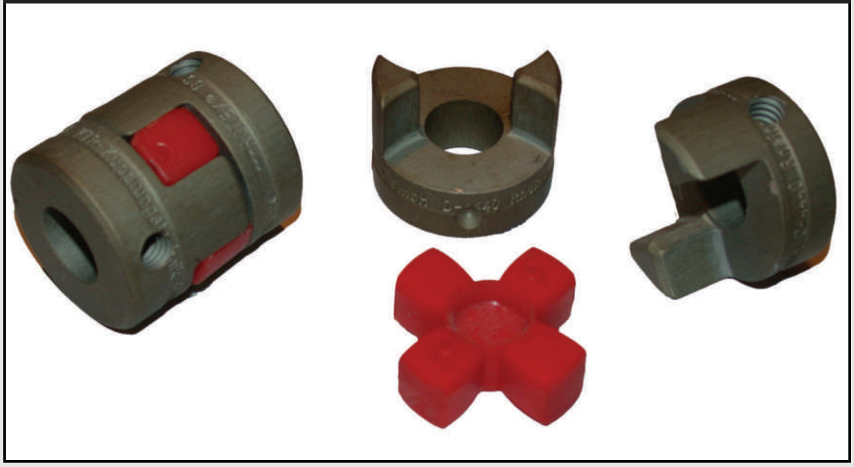


Figure 1. Assembled and disassembled spider couplings.

A. I would suggest that you take a look at go-kart web pages next. They have a lot of spare/replacement parts and after-market parts for a wide range of go-karts — simple street recreational karts, off-road karts, and even professional racing karts. Other places to look are vendors for golf carts, three- and four-wheel all terrain vehicles (ATVs) sized for children, and even power lawn mowers. For some low-cost parts, take a look at your local lawn mower or small engine repair shop. There, you can find some great deals for powered vehicles/motors, since a lot of them end up being abandoned because the original owners didn't want to pay their repair bills.

One online place to look for a differential is Kart World (**www.kartworld.com**). They sell a Comet differential for $199.00 (US). This differential weighs only seven pounds and works with drive systems up to 14 Hp.

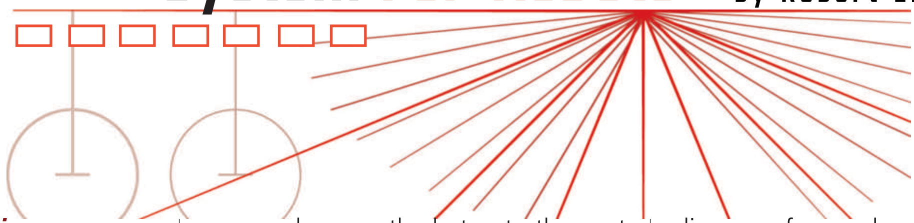
The drawback to this differential is that it has 1 inch diameter output shafts instead of the 1/2 inch shafts that you are looking for. I don't see that this should be a problem, though. You can either use the 1 inch shafts in your robot or you can cut the ends of the shafts off so that you can mount it to your 1/2 inch drive components. Then, you can use a spider coupling between your 1/2 inch diameter shafts and the 1 inch diameter axles from the differential. Spider couplings are nice, since they can tolerate some shaft misalignment. Figure 1 shows a photograph of a spider coupling.

McMaster Carr (**www.mcmaster.com**) sells a wide variety of these couplings. For example, P/N 6408K15 has bore diameter options from 1/2 to 1-1/8 inches. Select one with a 1/2 inch bore and a second one with a 1 inch bore. They cost about $8.29 (US) each. You will need two to make a set. Then, choose the spider material, such as Buna-N (P/N 6408K75, $3.99) or Hytrel (P/N 6408K95, $12.10).

The Hytrel can handle higher torque ratings than the Buna-N, but it can be harder to assemble. The go-kart industry is one of the most popular sources for larger robot parts, such as those commonly found in combat robots, FIRST robots, and outdoor industrial robots. Hopefully, you will be able to find what you are looking for at these places. **SV**

A Flexible, Low Cost, Infrared Object Detection System For Robots

by Robert E. LeDoux



This article discusses an infrared (IR) obstacle detection system for robots. The infrared sensor used is the Sharp GP2Y0D340K, which costs about $8.00. Controlling the sensor is a PIC 12C508A or 12C509A processor, which retails for about $1.60. Each sensing unit consists of the Sharp sensor, PIC, and a few passive components. I call this sensing unit a PIC/sensor pair or simply a sensor pair. This system is flexible because any reasonable number of sensor pairs can be linked together.

This system fits perfectly into a robot designed around subsumption architecture (see the Reference box). The IR sensor system does its thing — latching its findings onto output pins without creating processing overhead. Other robot processors can then interrogate the sensor findings, as they require.

The sensor system works much like a set of runners in a relay race. On the start signal, the first competitor runs the course and passes the baton to the next runner. That competitor runs and passes the baton to the next runner; the process continues until the race is finished. In our sensor system, each sensor pair takes its turn to look, then passes the turn to the next sensor pair down the line.

Figures 1 and 2 show some different sensor pair configurations. The minimum configuration consists of one PIC/sensor pair. Figure 1 shows a sensor string, initiated by an outside command. This string makes one pass through the line of sensors, then stops and waits for the next start command. Figure 2 shows a self-activating sensor ring. It self-starts and operates in a continuous manner. Details about wiring up these configurations are given later.

The PIC Program

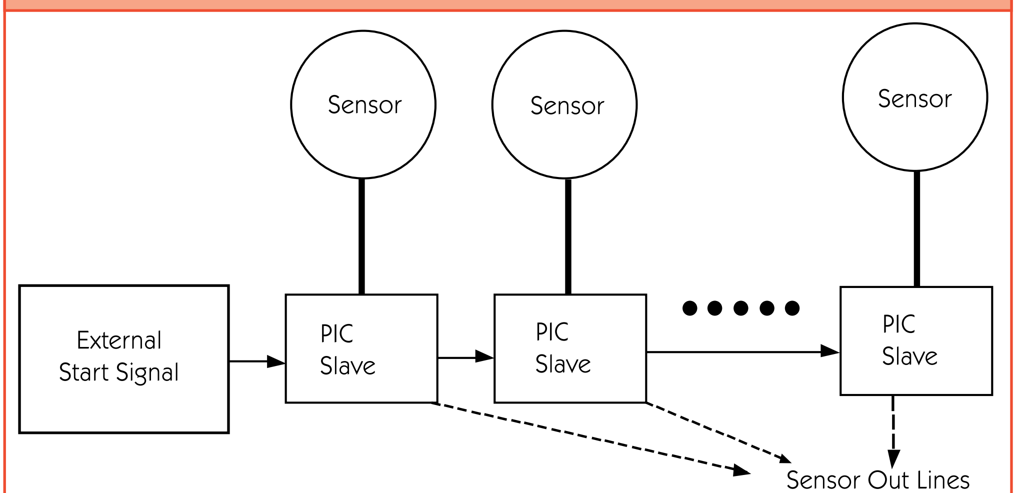
The assembly source code for the PIC is available for download from the *SERVO Magazine* website (**www.servomagazine.com**). To understand the sensing process, let's look at the functions from the viewpoint of one PIC. Figure 3 displays a schematic diagram for each sensor pair. We will reference the PIC pins in that diagram.

Step 1: Pin GP0 (pin 7) is the "input enable" line. When the previous PIC, or an external signal source, pulls this pin from high to low, our PIC takes its turn to process the sequence described in Steps 2 through 6.

Step 2: The PIC drives GP2 (pin 5) from low to high. This switches on the transistor that powers up the Sharp sensor. It takes about four milliseconds for the Sharp sensor to turn on and stabilize. The sensor begins taking a new reading every 6.4 milliseconds. The output of the Sharp sensor is read in on PIC GP1 (pin 6).

Sensing an object pulls the sensor line from high to low. The PIC is programmed to read the sensor output every eight milliseconds. If five readings in a group show a return, the readings come to an end. Otherwise, the PIC will make up to eight attempts to find low readings.

Figure 1. A sensor string, activated by external signal. Solid lines represent enable in/out connections.



Step 3: If five readings out of up to eight attempts show a return, the PIC latches the "sensor out" pin GP5 (pin 2) low. Otherwise, the "sensor out" pin is set high and maintains its state until a future sample process changes it.

Step 4: The PIC pulls GP2 (pin 5) back to low. This powers down the Sharp sensor.

Step 5: The PIC drives the "enable out" pin from high to low (GP4, pin 3). This pin remains low for one millisecond, then the pin returns to

Figure 2. A sensor ring that self-initiates and operates continuously.

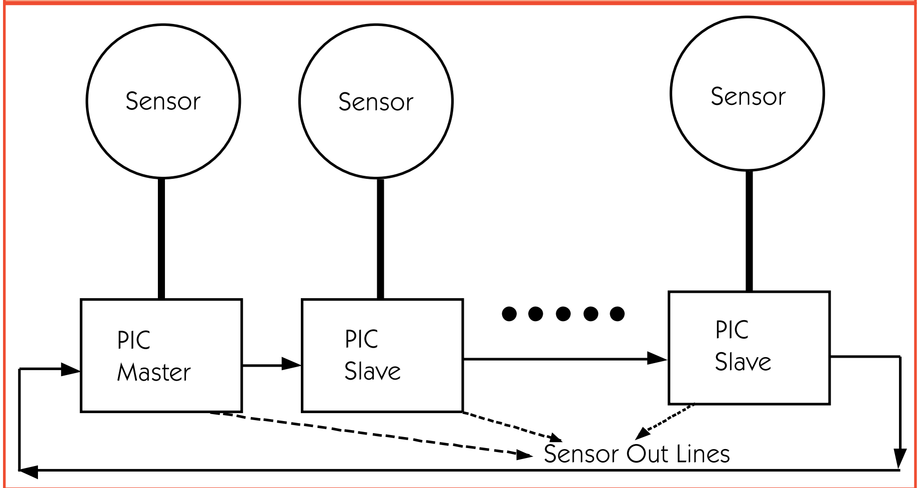
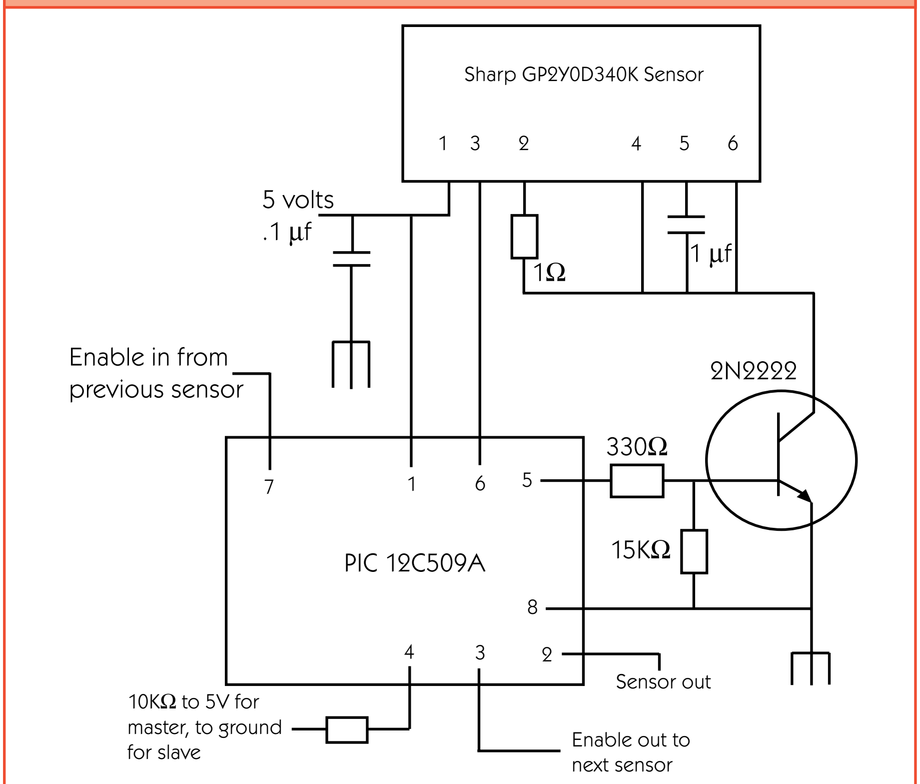


Figure 3. Schematic of a sensor pair.



high. This pin is connected to the next PIC in the chain as the "input enable" pin.

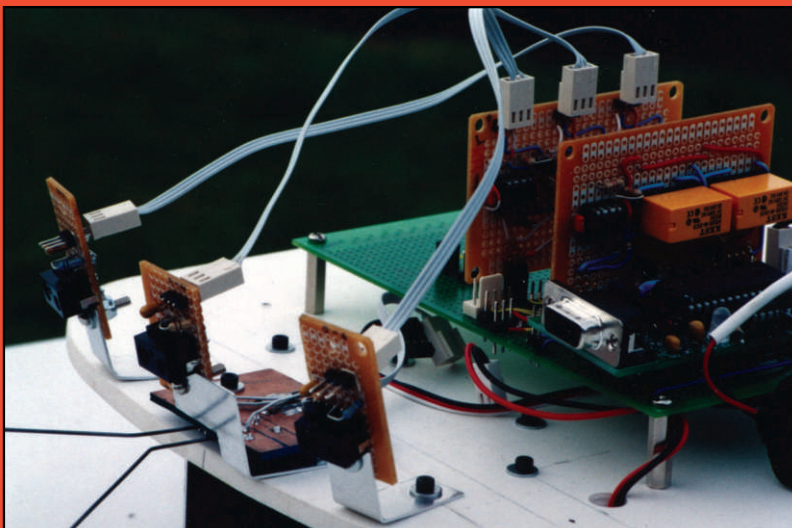
Step 6: The next PIC in the sensor pair series starts at Step 1, above. In this manner, each PIC takes its turn and then passes the turn to the next PIC in the string. Each turn should take about 70 milliseconds. Pin 2 (GP5) is the "sensor out" pin. Each PIC has a GP5 pin reporting what that sensor pair saw.

It is your responsibility as the robot designer to process this data. One possibility is to allocate one input pin in the robot's central processor to read each PIC.

Looking at the Sharp GP2Y0D340K Sensor

This sensor is a recent innovation from Sharp. Many robot builders have been using other Sharp GP2 series IR sensors, like the GP2D12. Most of these sensors are intended to be mounted with screws onto a flat surface. Electrical connections are made

Figure 4. The Explorerbot has three IR sensors. The PICs are mounted on the back vertical circuit board. On this robot, individual circuit boards stand on a backplane. The front vertical board has two relays to control the drive motor and a PIC to control the steering servo.



using three or four pin JST connectors.

The Sharp GP2Y0D340K used in this project, however, is a different animal. It is a bit less expensive and does not require a specialized JST connector because it is soldered directly to a circuit board. The sensor requires two external components: a resistor and capacitor. If these two components are mounted on the same board as the sensor, then three electrical connections will be required for operation: five volts, ground, and "sensor signal out".

The Sharp's "sensor signal out" line stays high unless there is an IR return, in which case the line is pulled low. The sensors are factory set to report an IR return when an object is found within 40 centimeters (about 15 inches).

While the Sharp sensor is rated for an average current draw of about 28 milliamps, it pulls an instantaneous 300 milliamps when the sensor's LED fires. Multiple sensor operations need to be managed to keep current draw under control and ensure that each sensor reads only its own LED output.

This design only allows one sensor to operate at a time. This limits the maximum current draw, as only one LED can fire at a time. The sensor power routine also ensures that each sensor only reads its own LED. The power routine employs a power-switching transistor with each PIC/sensor pair. A 2N2222 or 2N3904 and two resistors make up the switching circuit. This transistor applies power to a sensor only when it is that sensor's turn to operate.

A PIC is Wired as Master or Slave

While all the PICs are programmed identically, a PIC can be wired as a "master" or a "slave".

The PIC is wired as a master when that PIC is chosen to initiate processing within a sensor chain. A PIC is made master by attaching pin 4 (GP3) to five volts through a 10K resistor. A master works by bringing its own "input enable" pin low to start the ring moving.

This causes the master PIC to perform its sensing process, then pass the turn to the next PIC in the series. After the first pass, the master PIC operates like a slave.

There will be no more than one master PIC in a sensor pair chain. All the other PICs are wired as slaves. A PIC is made slave by attaching pin 4 (GP3) to ground through a 10K resistor. A slave is distinguished from the master in that it will not function unless its "input enable" pin is pulled low by an outside source.

Sensor Configurations

Configuration 1: Sensors can be wired as a sensor string. This layout employs a string of PIC/sensor pairs, as shown in Figure 1. The command to sense arrives from an external source, such as the main robot microprocessor. That command comes on the "enable in" line to the first PIC in the string. The PIC begins sensing when this line is pulled from high to low.

The first PIC passes the turn forward until the last sensor pair finishes. The sensor line then waits until the external command is again given to the first PIC in the string. All the while, the sensor results remain latched on each PIC's "sensor out" pin.

To setup this arrangement, wire each PIC as a slave. Wire the external signal source to the first PIC's "enable in" pin. The "enable out" pin on the first PIC should be wired to the "enable in" pin of the second PIC. Continually chain each PIC together in this manner. Wire the "enable out" pin of the last PIC in the line to five volts through a 10K resistor.

If you want a single sensor pair to operate under external command, wire the PIC up as a slave. Wire the "enable in" pin of the PIC to the external signal source. Wire the "enable out" pin to five volts through a 10K resistor.

Configuration 2: The sensors can be wired as an autonomous sensor chain. This layout creates a self-starting, continuously operating ring, as shown in Figure 2. This is like the previous setup, except that there is no external signal source. Wire one PIC as a master and the remaining PICs as slaves. The "enable out" pin of each PIC is wired to the "enable in" pin of each succeeding PIC.

Close the chain into a loop by wiring the last PIC's "enable out" pin to the first PIC's "enable in" pin.

Finally, a single sensor pair can be wired to operate independently. Wire the PIC as a master. Pull the "enable in" pin to ground through a 10K resistor. Pull the "enable out" pin to five volts using a 10K resistor.

Circuit Board Arrangements

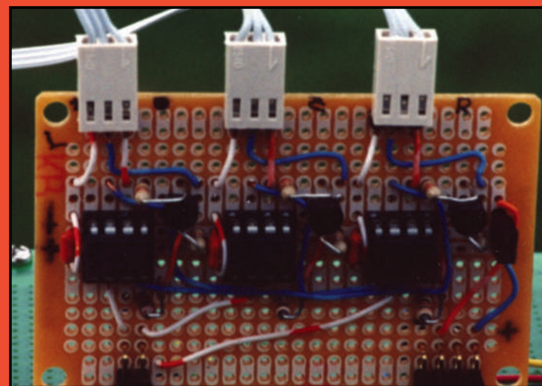
Arrange your circuit boards to best fit your application. As my robot application only required three sensor pairs, I chose to mount the three PICs and switching transistors on a common board (Figure 4). Each Sharp sensor, with associated resistor and capacitor, was mounted on a separate remote board (Figure 5). The disadvantage of this approach is limited flexibility.

My PIC circuit board has no expansion capabilities. I have to wire up an additional board to add additional sensors. Another option is to wire each sensor and its PIC on an individual board. This provides greater flexibility, as additional sensors can be added by simply plugging them into the chain or string.

Each board could have a three-pin input header and a three-pin output header. The input header would carry five volts, ground, and the "input enable" line.

The output header would pass on five volts, ground, and carry the "output enable" line. One additional line, the “sensor out”, would be required from each board to the robot central processor.

Figure 5. The PIC board is a RadioShack 276-150 proto board. The bottom right header is the power supply and bottom left carries the PIC "sensor out" data.



Final Details and Other Ideas

The assembly source code contains

even more details. Weak pullups have been activated to bring input lines high.

Note that all six GPIO pins are used. The program is not very long, nor does it employ any esoteric routines. My three-sensor system is reliable, but I recognize that a chain-type system is susceptible to outside noise.

As the links become more complex, there is an increasing chance for system lockup. I set the watch dog timer (WDT) prescaler to a ratio of 1:128 and, thus, it triggers about every second. When you burn a program into the PIC, you can decide whether or not to activate the WDT. For my continuous chain arrangement, I chose to activate the WDT.

Just in case a PIC locks up, the WDT will reset it and allow the processing to continue. Otherwise, one PIC crashing would bring the chain to a halt.

In a string arrangement, like the one in Figure 1, I would not activate the WDT. As the string depends on an outside signal source, WDTs could reset the PIC if it takes more than one second between each sensor command. If you want to try experimenting, modify the system to eliminate the input enable pin and replace it with the PIC master clear (MCLR). This could simplify the processing loop in the PIC and would cause each output enable line to reset the next PIC in the string.

This would also slow down the sensing process, but might make the sensor line less susceptible to external noise. Good current flow to the sensor is essential for proper operation. Operating properly, the sensor should pull its signal line low when it senses an object at a distance of about 15-16 inches. If sensing only takes place at a closer distance, then check the size of the power lines to the sensor. My sensors are connected to the main PIC board using six inches of #28 wire. My sensors will detect an object at about 14 inches. When I tested my arrangement with #30 wire wrap, the sensors were only good to about 11 inches. The base resistor on the switching transistor is 330Ω. Heavy base current flow is necessary to keep the transistor saturated. If you raise the value of the base resistor, you may find that the sensor fails to operate.

This PIC/sensor system replaces the scanning IR system on my robot, Explorerbot. My scanning system employed a stepper motor to swing an IR sensor back and forth in the forward direction.

Even when a dedicated UCN5804 chip was used to drive the stepper, the process required considerable processing resources. The time it took to perform a scan and process the data made the system marginally effective.

Too often, the front whiskers performed the object detection prior to the IR sensor, but, by moving to this multiple sensor system, my robot became aware of obstacles in a more timely manner. **SV**

Components List

Quantity	Part	Source and Part Number
1	12C509A PIC	Digikey-12C509A-04/P-ND
1	GP2Y0D340K sensor	Digikey-425-1810-ND
1	1Ω resistor	Digikey-P1.0BACT-ND (Pkg. of 10)
1	10KΩ resistor	RadioShack 271-1335 (Pkg. of 5)
1	330Ω resistor	RadioShack 271-1315 (Pkg. of 5)
1	15KΩ resistor	RadioShack 271-1337 (Pkg. of 5)
1	1 µF capacitor	RadioShack 272-1055
1	0.1 µF capacitor	RadioShack 272-109 (Pkg. of 5)
1	2N2222 transistor	RadioShack 276-2009

Reference

Rod Brooks' original article on subsumption architecture can be found at this website:
ftp://publications.ai.mit.edu/aipublications/pdf/AIM-864.pdf

CUTTING EDGE ROBOTICS

A Multi Function Robot — Part 3: Peg-Leg

by John Myszkowski

Ahoy, mates, ahrr, and the other things.

I know you've all been waiting anxiously for this installment. Well, the day has come; here we are together again, so let's have some fun.

Make It Walk

In the past, *SERVO* and *Nuts & Volts Magazines* have published articles on various types of walking robots. We are going to build on that knowledge and modify our mini-sumo for walking.

There are many ways to make a robot walk. A bipedal walking robot — a robot with two legs — seems almost ideal, but for that, you would need to use a motor for every joint. Even though you would end up with a pretty impressive-looking walker, it would also be very expensive and complex.

On the cheap side of the walking robot world is the cam and linkage drive. This type utilizes the existing motors to provide the motive power by converting rotary motion into linear motion. Even this type of technology has many possible solutions. The photos show you a few examples, but we will stick with the one that is best suited to our chassis (Figure 1).

The Cutting Edge multi-function robot has been designed to be modified and expanded. The chassis — or base — is designed to be simple and easily reconfigured by utilizing the extra holes. The linkage system I chose for this project is a crank and follower type. It is similar to a cam and cam follower, but we have a crank in place of the cam.

A crank, lever, and slide system (crank follower and the middle leg) will replace the wheel. The motor will drive a crank, which, in turn, will move the legs. If this sounds complex, it's because it is. To make it seem less complicated, you can spread out the project and make each part separately, one at a time.

The Creative Edge

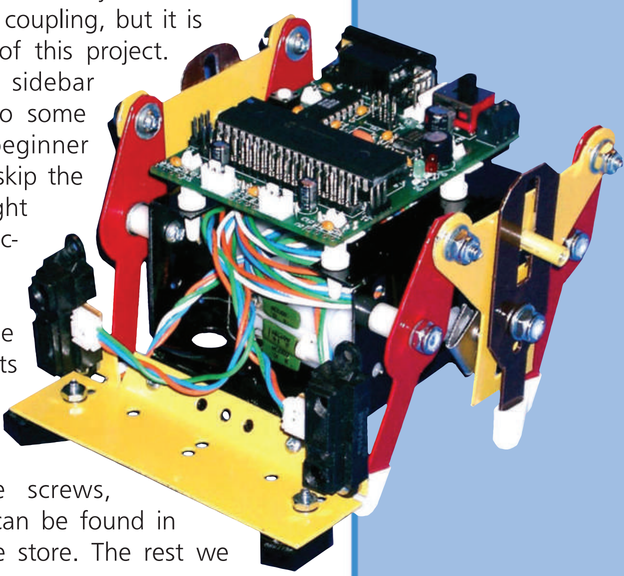
Before we begin the actual fabrication, I would like to mention a couple of things which may have some impact on your project building. Keep in mind, the Cutting Edge projects are meant to be fully functional and also educational. They are meant to present a series of technical exercises, which will, hopefully, lead you to a better understanding of robotics. The electro-mechanical principles presented here can be scaled up or down and applied to small or large robots. The series will provide you with enough background experience and self-confidence to tackle any robotics projects you may have planned for in the future.

Remember, exercise your creativity and:

- Do not be afraid to experiment with the given designs and templates.
- Do change the dimensions.
- Do not be satisfied with your finished project, whether it works or not.
- Do modify your project and add to it.

I could go into the theory behind motion transfer and power coupling, but it is beyond the scope of this project. The references sidebar should point you to some relevant, beginner resources. We will skip the theory and hop right over to the construction details.

We will need our sumo from the previous installments of "Cutting Edge Robotics" and some mounting hardware. All the screws, spacers, and nuts can be found in your local hardware store. The rest we



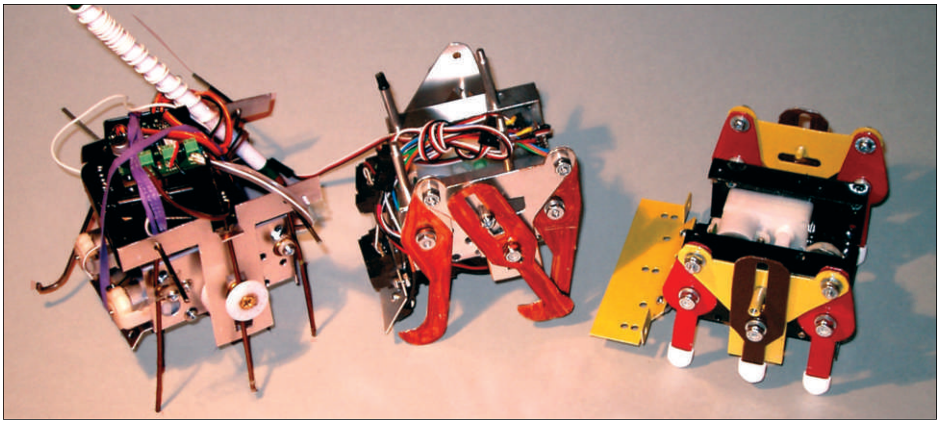


Figure 1. Left to right: Peg-Leg-1 (prototype), Peg-Leg-2 (Tipsy), Peg-Leg-3 (Peg-Leg).

will need to fabricate (or purchase from **www.novarobotics.com**).

First Things First

Take off the wheels and take a look at the underside of the chassis. As you can see, there are a number of unused holes that are just crying out to us, "Use me! Use me!" (Figure 2). All those extra holes are perfect for adding some legs. You have a choice of "Peg-Leg" or the less stable "Tipsy" leg sets; you can even make both for experimentation (Figure 3).

The complete set of legs for each side of the robot is made up of five small pieces of sheet aluminum (Figure 4). You may choose thick sheet plastic to make the pieces if you prefer, but aluminum is much more flexible. I chose a thin (about 1 mm) aluminum sheet for my projects because of its durability, relative stiffness, and ease of processing. It is also available from most metal suppliers, such as Hobby Engineering (**www.hobbyengineering.com**).

Do keep in mind that it may be more difficult to finish your robot if the material you choose is not the one I suggest.

Thinner or thicker material can be used, but a higher skill level may be required for building the parts. Thicker sheet aluminum (2 mm) will be more rigid, giving better performance. Thinner stock will be easier to cut and bend, but will require extra support to give it more rigidity.

I have provided templates for creating two different sets of legs. The same parts can be rearranged and used for two different styles of motion conversion. You will need to obtain a 100% (1:1) printout of the templates before you can begin. The files are located on *SERVO Magazine*'s site, in the FTP Library (**www.servomagazine.com**).

A Safety Reminder

Think "safety"; work safely. Some people like to use thick leather gloves to hold the work piece while drilling. For many, it feels safer, but keep in mind that it may be more of a hazard than a help; gloves can get tangled in the moving machinery. Think twice about safe ways to work before you start. Thin latex or non-latex gloves keep your hands clean and provide a non-slip grip.

Keep your hands away from the path of a moving tool or object.

Now It Begins

Print out the templates first (Figure 5). Glue the paper templates to your sheet metal. If you are not sure how, see Part 1 of the Cutting Edge series for details.

Poking Fun

Drill all of the holes in all of the pieces before cutting them out. The parts are pretty small and hard to handle, but, if you keep them together on one piece of sheet metal during fabrication, then handling and drilling will be easier and safer.

Hold down your work as you drill to prevent it from going for a spin and damaging your hands.

The two small cranks that move

Figure 2. Now, we find out what some of those extra holes are for.

Figure 3. Peg-Leg is the better performance robot, but Tipsy is good for experimentation. The legs can always be made shorter.

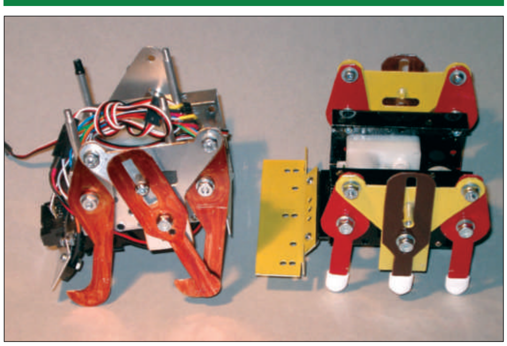


Figure 4. Two sets of legs total 10 pieces. It's a lot of work, but worth it.

Possible Problems and Solutions

DC Motors

DC motors generate a large amount of electromagnetic noise. This is usually the biggest problem with autonomous robots. You may notice that the behavior of your robot is weird, it doesn't do what it should, or the micro fails to reset properly.

Most likely, this problem may be caused by voltage pulses induced within the electric motor. Motor windings are inductors; they generate extra voltage when power is first applied or taken away and when they move within a magnetic field. The average potential in a small DC motor can be as high as 20 volts; the peaks can be much higher. The most troublesome aspect of the noise is the generation of negative spikes within the windings. All that extra noise is often high enough to eventually latch up any micro. If the circuit is not sufficiently protected, then the simple act of turning the robot wheels by hand can burn out some of the electronics.

However, this generated noise isn't *all* that bad. You can actually utilize this power generating ability of our little gear motors to create a variety of crank powered items. In this application, the ability to generate power is a hindrance, rather than a help.

Three small capacitors (0.01 to 0.1 µF) for each motor are all you need to clean up the unwanted noise. One capacitor is soldered between each terminal and the motor case and one capacitor between the two terminals (Figure A). The capacitors act like short circuits to the fast, high frequency voltage pulses, which stop before they have a chance to go any further. The slower pulses do get past the capacitors, but they get trapped by the built-in diodes within the H-bridge (motor driver circuit).

Sharp GP2D Sensors

If you get a chance, read Sharp's specifications for their GP2Dxx sensors. They flatly state that these sensors *cannot* be used in any sensitive or critical applications; they are great for hobby or experimental use, but have some serious reliability problems. There are three well-known difficulties with these sensors and none of them is very well documented.

The first difficulty is of an electrical nature; the sensors are prone to electromagnetic noise pick up. This is more evident with analog versions, but even the digital versions become erratic at extreme distances.

The second problem is also electrical, but it is the reverse of the first; these sensors generate a lot of noise and consume a relatively large amount of power. This noise is due to the switching (modulating) of power to the diode at high speeds. A couple of simple things can reduce these two problems and increase the sensor's reliability.

One suggestion is to place a capacitor (about 10 µF) across the two power leads. The closer the capacitor is located to the sensor, the better. I would even go as far as using two capacitors: one large (10 µF) and one small (0.01 µF) for the higher frequency spikes. This is not a necessity and can be considered a last resort.

A partial solution is provided by the manufacturer; they made the case from electrically conductive plastic, which acts like a shield when grounded, so we simply need to complete the circuit. Our chassis is metal, so we need to connect the negative (minus) side of the power supply to it. Some paint can be removed for better conductivity or you can connect the power ground directly to the sensor mounting screw. You will have to use an ohmmeter to check for electrical continuity between the plastic sensor case and the minus side of the power supply.

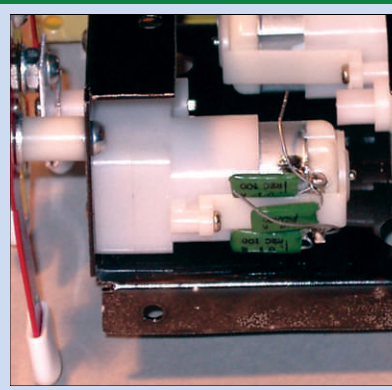
The third problem is of a more physical or mechanical nature and has the potential of becoming an electrical problem, causing sensor failure. By design or omission, the connector leads were not bent over before being soldered, leaving the connector vulnerable to weak solder joints. This leaves the whole connector and harness hanging only by the strength of the solder joint; unless corrected, you are going to end up with an eventual electrical and mechanical solder failure, which will result in erratic or failed operation.

To secure it, the connector should be re-soldered and glued to the circuit board. After plugging in the harness or cable, secure it with a cable tie or a dab of hot glue. A different solution is to remove the connector altogether and solder the wire harness directly to the circuit board. This is a good way to eliminate the need to buy special mating connectors for the Sharp sensors, but you will need to make sure the ends of the wires are glued to the circuit board.

Other Solutions

There are other small problems that are sure to come up. Feel free to share them, as well as your solutions, with others. A good place to meet with other Cutting Edge builders is through the Cutting Edge Projects Yahoo group: **http://groups.yahoo.com/group/CuttingEdgeProjects** Use it for any questions and for displaying your own finished (or unfinished) robots. I will post all of the extra details, as well as user additions and software. The support and information website for all the Cutting Edge Projects is at **www.cuttingedgeprojects.com**

Figure A. The three noise suppression capacitors are shown here. They are a standard procedure to make electric motors less noisy.



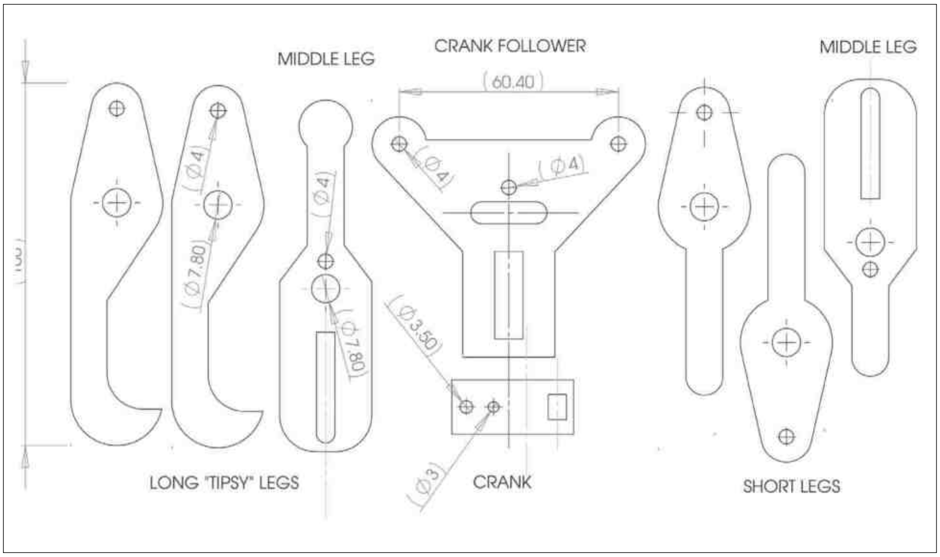


Figure 5. This is the template for the legs. The cranks and crank followers are the same for Peg-Leg and Tipsy.

the crank followers and middle legs need a rectangular hole. With a little practice, you will be able to make them quite easily — it only seems difficult at first (Figure 6).

First, drill two small holes inside the indicated rectangle (Figure 7). Use a small pair of side cutters to remove as much excess material from the inside of the hole as you can (Figure 8). Use a small square or flat file to enlarge and finish shaping the hole.

Keep one of the motors handy so you can fit the shaft into the hole as you are enlarging it. The crank will work best if the hole is not too sloppy (Figure 9).

While drilling, filing, or nibbling out the rectangular holes, you have to make sure that the template paper is secure. It is the most relevant reference that you have for cutting out parts and bending them.

The long, rectangular holes in the sliding parts need to be smooth and snug, but not tight. Keep a washer (one that will slide within the hole) handy while finishing the holes.

It is easy to make the holes too sloppy, but they should still operate correctly. The second set of legs will be easier to make and look neater; you may also want to make a third set of legs, just for practice (Figure 10).

Slice and Dice

Cut out all of the pieces. You can use hand tools, like sheet metal snips, but I suggest you use a band saw, if you have one. There are a lot of little pieces and it will take some time to cut everything out. Take a break if you feel tired — there will be more waiting for you after the break (Figure 11).

Note: Now is a good time to de-burr and clean all sharp edges.

The only thing that needs bending is the crank itself (Figure 12). Follow the photos closely to get an idea of what needs to be done. There is no need to bend anything else, unless you want to do so for cosmetic reasons.

The Gathering

Now that all the dirty work is done, you are ready to assemble the components and complete Peg-Leg, the autonomous sumo walker. Check the assembly diagrams for the order of assembly and details.

Getting Cranky

Put together the crank assembly first (Figure 13). Thread the small hole in the crank follower, matching the thread on the metal stud. This can be

Figure 6. The two cranks make it all possible — pretty simple, yet sophisticated.

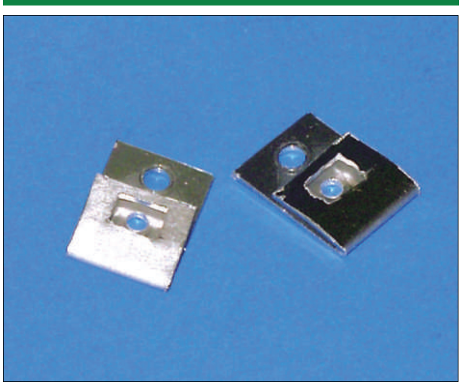


Figure 7. Make the slots by drilling two (or more) small holes first.

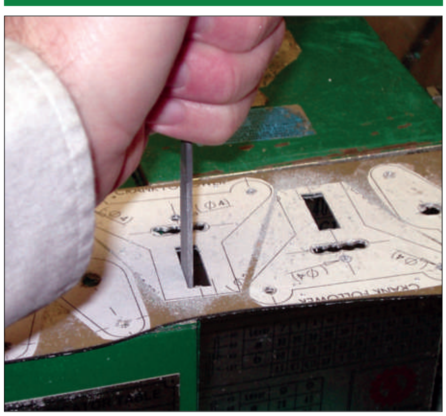
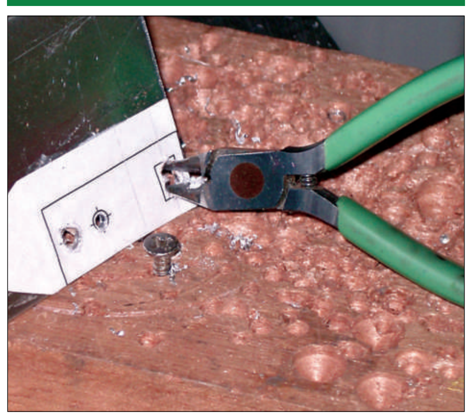


Figure 8. To make room for the file, remove as much excess material as possible.



accomplished with a proper tapping tool or simply by forcing a steel screw of the same size as the stud into the hole. Since the material is aluminum, the metal stud will make its own thread.

If you haven't already done so, screw in the stud now. Insert the 1/2" screw into the crank and tightly fasten it with a nut. The flat metal washer goes on the screw next, followed by one plastic washer. Place the vertical slot of the crank follower over the plastic washer, followed by one metal washer and one plastic washer.

Place the middle leg over the plastic washer, followed by a metal washer, and then the locknut. Tighten the assembly screw now, but make sure the crank follower and the middle leg move freely on the plastic washers. Insert the rectangular hole of the crank over the motor shaft.

Finally, attach the crank to the motor shaft with the same small screw which originally fastened the wheel. The small hole in the middle leg is for accessing the small screw.

Great Legs

The outside (front and back) legs need to be attached to the robot chassis next (Figure 14).

Use the drawings as a guide and try to keep the legs, crank, and follower parallel to each other. Some motor shafts may be longer than others, causing the legs to get tangled if not parallel. Finally, attach the legs to the crank follower (Figure 15).

Again, make sure everything is parallel in order to prevent the legs from getting tangled. If you use too many spacers, then you may need to get longer screws. If, for some reason, you find that keeping the pieces parallel is impossible or difficult, then you can bend the legs. It's an easy way out, but it works, too.

Make Them Happy

Paint and decorate your robot according to your own tastes. A little bit of paint can turn a drab clunker into a fierce and exciting little battle monster. The paint can always be removed or covered with a different color. Stickers or glued-on decorative pieces will add to the robot's personality and make the audience happy to see it.

References

BOOKS:

Building Robot Drive Trains
ISBN 0-07-140850-9

Build Your Own Robot
ISBN 1-566881-102-0

Junkbots, Bugbots, & Bots on Wheels
ISBN 0-07-222601-3

Check out the following internet links. They are all related to the Cutting Edge projects.

SUPPLIER AND INFORMATION WEB LINKS:

www.servomagazine.com
www.novarobotics.com
www.hobbyengineering.com
www.1sorc.com
www.basicx.com
www.cuttingedgeprojects.com
www.robotgames.ca
www.robotgames.com
www.solarbotics.com
www.barello.net
www.bugnbots.com
www.junun.org
www.avrfreaks.net
www.ridgesoft.com

YAHOO USER GROUPS:

http://groups.yahoo.com/group/CuttingEdgeProjects

http://groups.yahoo.com/group/MiniSumoMarkIII/

http://groups.yahoo.com/group/Sumo11users/

http://groups.yahoo.com/group/tabrobotkit

Figure 9. Keep the motor handy to try the crank for size.

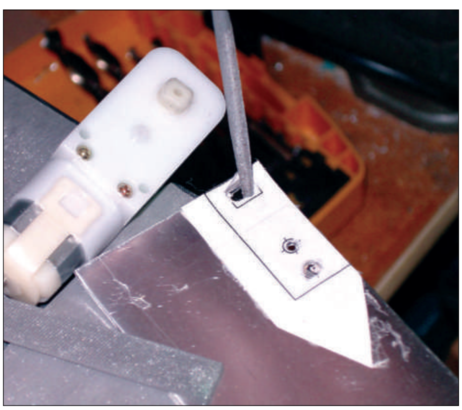
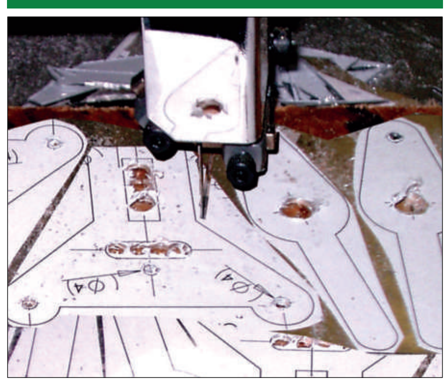


Figure 10. While making the rectangular holes, keep sizing them with the matching part. This will reduce the chance of sloppy parts.

Figure 11. You can certainly use hand snips to cut out the pieces, but the work goes much faster with a band saw. A band saw is one of the better tools in a home shop.



Peg-Leg Parts List

- **1** — crank
- **1** — crank follower
- **1** — middle leg (Peg-Leg or Tipsy size)
- **2** — legs (Peg-Leg or Tipsy size)
- **1** — threaded stud (3/16" diameter, at least 1/4" long)
- **1** — screw (6-32, 1/2")
- **4** — screws (6-32, 3/4")
- **9** — flat metal washers (#6 — larger diameter than the plastic spacers)
- **6** — plastic washers, spacers (5/16" and thicker than the crank and follower material's thickness)
- **2** — plastic spacers (5/16" diameter, 3/8" long and padded with flat washers for best fit)
- **1** — nut (6-32)
- **5** — locknuts (6-32)

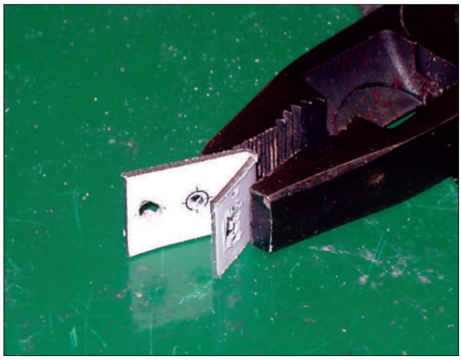


Figure 12. The crank can be folded with a pair of pliers, but practice on a piece of scrap metal first.

Contemplate Your Options

If you are careful in your assembly, your walker will fit within the autonomous mini-sumo size restrictions. Competition may not be your goal, so you may consider other options. One of the fun things about robotics competition is being able to demonstrate your individuality. Dress up your robot and give it a name. A Peg-Leg model will be unique in the sumo ring just the way it is, but you can make it even more exceptional by adding personal touches.

A simple flip of the scoop will convert the sumo into a wandering explorer robot without rewriting the sumo program (Figure 16). Take off the scoop and mount it, as shown in the photo. The robot will move toward any object in its path, but then back away and turn when the line sensors get close enough to its target.

The mechanical functions can also be enhanced or modified by swapping or rearranging leg components. As shown in the various photos, we have more than one alternative for using the crank and cam idea. Power transfer and length of stride are two variables which will give you four different configuration options. With the longer legs, you can build Tipsy, which is the more unstable of the two, but fun to operate and watch, nonetheless (Figure 17).

The stride and gait can be modified by changing the length of the legs. The shape of the legs or the crank follower will also change things enough to give you a great opportunity to experiment with power transfer ratios and methods. Changing circular motion to linear motion is very interesting in itself, but even more fun to play with.

I chose to use sheet aluminum as a raw material, but you may want to use coat hanger wire, #12 or #10 gauge solid copper wire, plastic, or even wood (Figure 18). The key here is learning, not spending money — at least not a lot of cash. Attempt it my way first and — once you are happy with your creation — experiment, be creative, and have fun. If you think something will work, try it. If it doesn't work as planned, go at it again with a different idea. You can always build a new one if everything goes totally wrong.

Cautionary Note

As with all machinery, it is very

Figure 13. The sequence of components is suggested here, but you may apply your own wisdom for a better fit.

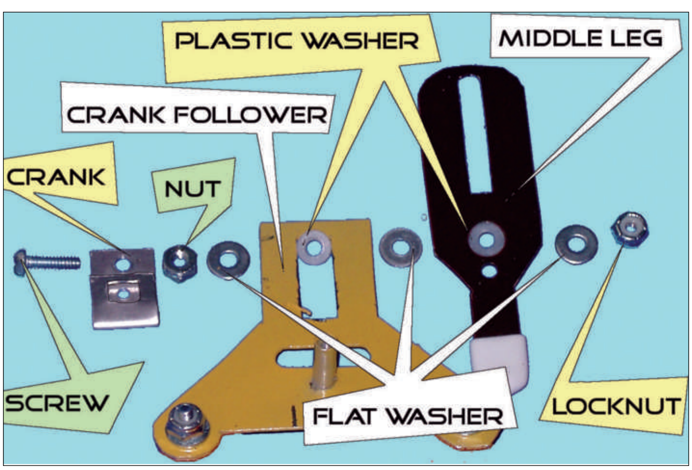
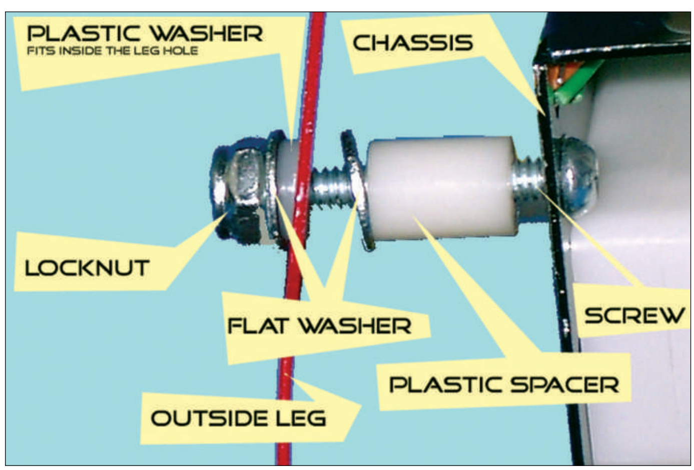


Figure 14. The sequence of components is only important if you want things to work on the first try.



unwise to stick your fingers into any moving parts. Peg-Leg is full of moving, sliding, and pinching parts. Treat it like any machine: with respect. If you have small children, then do not show the robot to them. They may cry with joy when they see it, but will cry awfully loud when their little fingers get caught between the moving legs.

It is unlikely that there is enough power in these small motors to cause damage to a hand, but the motors and the controller can be damaged if the legs get jammed for a long time. Let me repeat myself to emphasize the point ... please be careful.

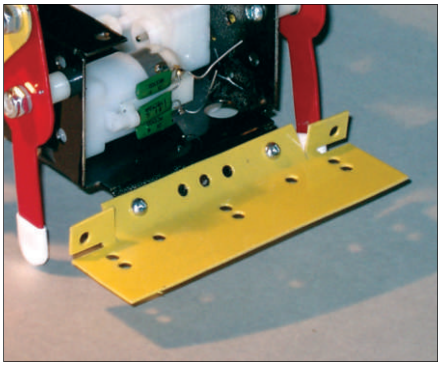


Figure 16. This photo shows how the scoop should look after re-mounting it. The sensors will fit on the vertical tabs and can be adjusted.

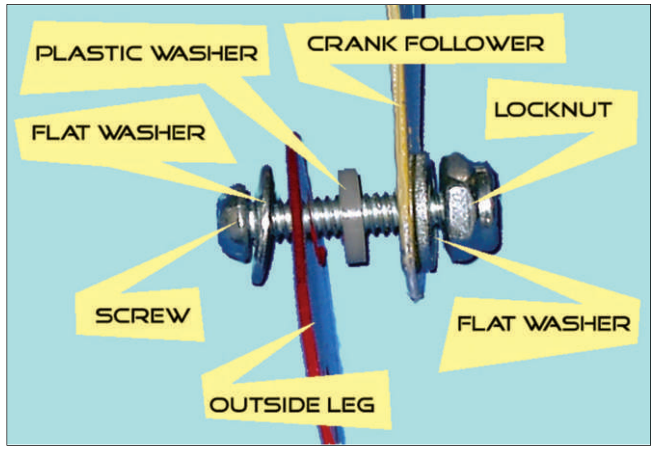


Figure 15. The order of the hinge parts has been pondered upon and a compromise has been reached. You are more than welcome to try your own sequence.

Controller and Software

The controllers mentioned in Part 2 can still be used and, from time to time, they will be referenced. The robot chassis and attachments, being mechanical in nature, have been specifically designed to work reasonably well with as many available controllers as possible. Within reason, these controllers will remain compatible with most, if not all, of the Cutting Edge projects.

A microprocessor-controlled robot is not much good without software, so I made sure that the walker configuration can use Chris Harriman's "simpleSumo" program for the BX-24 without program modification. Two cuts should be made in the scoop to

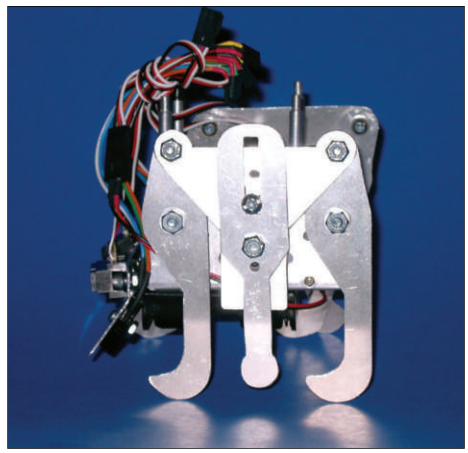


Figure 17. Tipsy is really unstable, but I think it is good to make the extra set of Tipsy legs to see what you can do to improve them. Hint: Making them shorter will help.

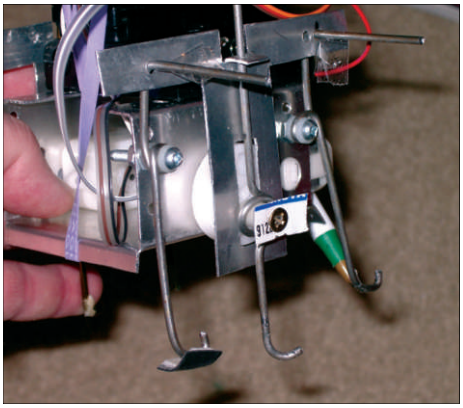
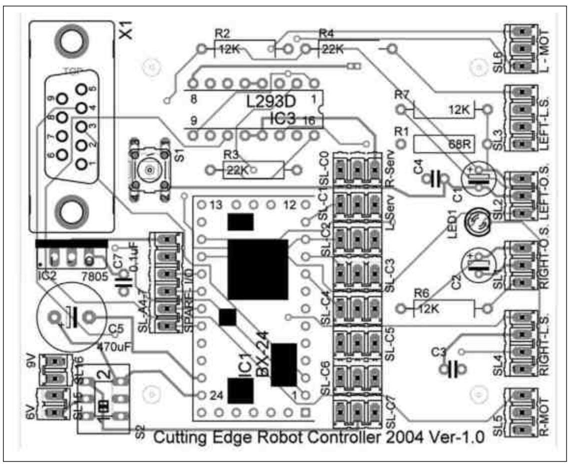


Figure 18. The prototype Peg-Leg was made with a piece of scrap aluminum and some coat hanger wire. Copper wire would have been even easier to work with. The crank was made from the plastic hub that originally came with the wheels.

make it simpler to reposition the Sharp object sensors (Figure 19). The "simpleSumo" software that Chris donated (thanks, Chris) can be found in *SERVO Magazine*'s FTP Library (**www.servomagazine.com**).

From this installment of the Cutting Edge series on, I will be using a "roll your own" prototype controller — the CERc (Cutting Edge Robotics controller), made with a NetMedia BX-24 controller module (Figure 20). Any single board controller can be used in this application, but the BX-24 has many features that will be needed in some of our future projects.

Figure 20. This is the layout of the CERc (Cutting Edge Robot controller) that will appear in next month's installment. It is BX-24 based and will run circles around any other Stamp-like device.



Let's Rap

By the time you see this in print, I will have a circuit board ready for your use. Check the *SERVO* FTP Library for updates and the Nova Robotics website (**www.novarobotics.com**) for components or kits for this project.

The plans and PCB layout for the CERc will be published and described in next month's construction article. **SV**

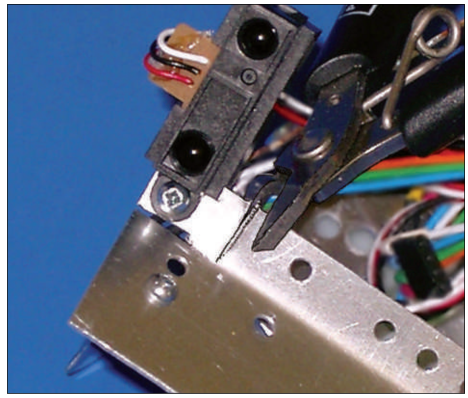


Figure 19. You can use a pair of side cutters to make a modification to the scoop. The cut will allow the object sensors to be adjusted in any direction.

About the Author

John Myszowski has worked in the area of electronics and mechanical design in the industrial automation and automotive industry for over 20 years and loves tinkering. He can be reached through Email at john@myszowski.net

MENAGERIE

Send us a high-res picture of your robot with a few descriptive sentences and we'll make you famous. Well, mostly. menagerie@servomagazine.com

Strider

Brett "Buzz" Dawson, Orlando, FL

Strider is a six-wheeled Mars exploration type robot which was built to test some ideas on articulated suspension and to show my high school students what you can do with a small budget, some hand tools, and a little bit of imagination.

All of the metal in this robot was purchased at surplus stores for a total of just under $7.00. The top of the robot is made from a pair of aluminum clipboards. The motors were found at a local electronics surplus store and the wheels are from a 1/10th scale R/C truck. Latex covered paddles on the front grasp objects with ease. The whole robot is teleoperated, but I plan to improve upon the design so that it can run autonomously without the fear of it twisting itself into a pretzel.

www.teamdavinci.com/robotstrider.htm

Buddy

Michael Misinonile, Clayton, NC

Our goal when building this robot was to use only parts already in our shop. It has a homebrew remote controller with nine separate control channels. It has two BASIC Stamp II modules and a Parallax modem system. The drive motors are modified Ford window motors. The head is run by a model airplane servo, and all other motors are just plain gearheads.

Buddy is used in my Show Bot rental business as an entertainment character.

FASTENERS FOR ROBOTS

by Gordon McComb

A robot is an amalgam of parts, both small and large, important and seemingly inconsequential. We tend not to overlook the big stuff: motors, wheels, batteries, and bases. It's easy, though, to forget how the robot is put together and, for most machines, that's with hardware.

Fasteners are the most elementary of all hardware. There are dozens of fastener types, so we'll concentrate just on those that are most practical for amateur robotics. These are machine screws, nuts, washers, and all-thread rod.

Machine screws are designed for fastening together parts of machinery, hence the name. Unlike a wood or metal screw, machine screws do not have a pointed end. The machine screw is designed to be secured into a nut or other threaded retainer. If the material being joined is threaded, the screw can be secured directly. Machine screws are sometimes referred to as bolts. When a nut (see below) is included, they're known as stove bolts. I like to limit the term "bolt" to a machine screw larger than 1/4" in diameter, though this is by no means universal. You're free to use any terminology you like.

Nuts are used with machine screws. The most common is the hex nut, so called because the nut has six sides. The nut is fastened using a wrench, hex nut driver, or pliers. While hex nuts are the most common and the cheapest, there are other types, including square nuts and T-nuts (also called blind nuts), for use with wood and soft plastics. Also handy in many robotics applications are locking nuts, which are like standard hex nuts, but with a nylon plastic insert. The nylon helps prevent the screw from working itself loose.

Washers act to spread out the compression force of a screw head or nut. Under load and without a washer, damage may occur because of the small contact of a machine screw head or nut. The washer doubles or even

triples the surface area, spreading out the force. Washers are available in various diameters to complement the size of any machine screw (or bolt, for larger sizes). Variations on the washer theme include tooth (internal or external) and split lock washers; these provide a locking action to help prevent the fastener from coming loose. To be effective, the locking washer must be correctly matched with the size of fastener it is used with.

All-thread rod is rod with common machine screw sizes and threads. In fact, it's commonly used when a specific length of machine screw is not available. All-thread is sold in one to six foot lengths. Apart from making your own custom-length screws, all-thread is good for making shafts and linear motion actuators.

Fastener Sizes

Fasteners are available in common sizes, either metric or standard. Standard-size (SAE) fasteners are denoted by both their diameter and the number of threads per inch. For example, a fastener with a thread size of 6-32 (also shown as 6/32) has a diameter referred to as #6, with 32 threads per inch (also called pitch). Diameters under 1/4" are indicated as a # (number) size; diameters 1/4" and larger can be denoted by number, but are more commonly indicated as a fractional measurement — 3/8", 7/16", and so on.

The pitch (number of threads per inch) can be either coarse or fine for standard fasteners. Therefore, not all #6 fasteners have 32 threads to the inch.

Metric fasteners don't use the same sizing nomenclature as their standard cousins. Screw sizes and pitches are defined by diameter, thread pitch (number of threads per millimeter), followed by length — all in millimeters. For example, M2-0.40 x 5 mm means the screw is 2 mm in diameter, has a pitch of 0.40 threads per millimeter, and has a length of 5 mm.

Anatomy of a Machine Screw

Machine screws have the following characteristics:

• **Head**. The head is used to drive the screw. Actually, not all screws have heads, but most do. Set screws are a good example of screws that don't have heads.

• **Shoulder**. The shoulder is the area of the screw right under the head. For most machine screws, the shoulder is threaded, but for other types of screw (like the wood screw or carriage bolt), the shoulder is plain or squared.

• **Shank**. The shank is the threaded portion of the screw. The shoulder and shank together determine the length of the screw (except on flat head screws). The diameter of the shank determines the size of the screw.

• **Thread**. All screws are, by their nature, threaded. The threads can be to the left or right; most screws use right-handed threads. The number of threads per inch or millimeter determines the pitch of the screw.

• **Point**. The typical machine screw has a blunt (or die) end, but other screw types have cone, cup, or taper points. Self-drilling screws may have a cone or "pinch" point, along with flutes that dig into the material. Set screws are usually pointed or cupped, depending on the application.

Fastener Head Styles

When buying machine screws, you have a choice among a variety of heads and drivers. The head greatly contributes to the amount of torque that can be applied to the screw when tightening it. Additionally, certain head styles are designed to have a lower profile, so they stick out less than others.

Round, pan, and flat head screws are, by far, the most common and they tend to be the least expensive.

• **Pan** heads are good, general purpose fasteners. However, the head is fairly shallow, so there is less grip for the driver.

• **Round** heads are taller and protrude more than pan heads, so they provide greater depth for the driver. They are good for higher torque applications, when it doesn't matter if the head sticks out.

• **Flat (or countersunk)** heads are required when the head must be flush with the material's surface.

• **Oval** heads are often used as a substitute for flat head screws, but when the head requires extra depth; oval heads are semi-rounded.

• **Button** types are much like round heads, but they are commonly used with specialty drives (see below, under Drive Styles).

• **Fillister** has an extra deep head for very high torque and the top of the head is rounded.

• **Binding** is a cross between pan and fillister; it provides a wide contact area and is ideally suited for making electrical connections to solid and stranded wire.

• **Hex bolt** heads use no slot and require a wrench to tighten them. They are used for highest torque applications.

Fastener Drive Styles

Most machine screws available at

the hardware store are slotted for flat-bladed screwdrivers. You may, instead, wish to use Phillips, Torx, square drive, Pozidriv, or hex head (Allen wrench) screws. All require the proper screwdriver. Slotted screws are cheaper to make, so they cost less, but there's a risk of stripping out the slot if the screw is over-tightened. Specialty drive screws can be tightened and loosened without as much risk of stripping out the head.

Here is an overview of several commonly available drive types and what makes them unique. For all types, different sizes of drivers are used to accommodate small and large fasteners. In general, the larger the fastener, the larger the driver. The larger the driver, the more torque can be applied for a sure fit.

• **Slotted** drives are made for general fastening and low torque drive; the screwdriver may slip from the slot.

• **Philips** have a cross-point drive, which resists drive slippage, but the head is easily stripped out when using an improperly sized driver.

• **Combination** drives work with both slotted and Phillips, in case you don't have the right screwdriver handy.

• **Hex, Torx, Pozidriv, and square** require a specific size and type of driver, which minimizes stripping. Hex drive is also called Allen, after the company that helped popularize this drive type. The disadvantage: You must have the proper tool to fasten and unfasten.

• **Socket** combines a fillister style head with a hex or Torx drive. The rim of the head may also be knurled, in order to assist in hand-tightening.

SELECTING COMMON FASTENER SIZES

Robot builders gravitate toward favorite materials — and fasteners are no exception. I can't tell you which sizes of fasteners to buy because your design choices may be different than mine, but I can tell you what is used the most in my robot workshop. Perhaps this will give you a starting point, if you're just now stocking up.

For small, tabletop robots:

I try to use 4-40 screws and nuts whenever possible because they're about half the weight of 6-32 screws and, of course, they're smaller. I use 4-40 x 1/2" screws and nuts to mount servos on brackets and 4-40 x 3/4" screws for mounting small motors. Larger motors (up to about six to eight ounces) can be fastened using 6-32 or 8-32 hardware.

For small robots, I try to keep the following fasteners in stock at all times:

- 6-32 nylon machine screws, in 1/2" and 1" lengths
- 8-32 x 1/2" and 1" nylon machine screws
- 4-40 steel machine screws in 1/2", 3/4", and 1" lengths
- 6-32 steel machine screws in 1/2", 1", 1 1/2", 2", and 3" lengths (I often use the 2" and 3" lengths to crease "risers" for robots with multiple decks.)
- 8-32 steel machine screws in 1/2" and 1" lengths
- 6-32 x 3/4" wood screws, when fastening together panels of rigid expanded PVC
- 6-32 x 1/4" blind nuts
- 8-32 x 1/4" blind nuts

For all sizes, in nylon and steel, I keep a corresponding stock of nuts, flat washers, and lock washers.

Note that nylon fasteners are not stocked at all hardware and home improvement stores and selection may be limited for the 4-40 size hardware. Mail order or hobby stores are two alternative sources for specialty fasteners.

Some tasks call out for even smaller fasteners than 4-40. I keep on hand a small assortment of 2-56 hardware, typically 2-56 x 1/2" and 3/4" screws and nuts. Hobby stores that cater to the radio control enthusiast are reliable sources for 2-56 fasteners.

For larger rover robots, I keep a stock of the following:

- 10-24 (not 10-32) steel machine screws in 1/2", 1", and 1 1/2" lengths
- 1/4"-20 steel machine screws in 1/2", 1", and 1 1/2" lengths

Larger sizes are purchased when needed. Specialty designs, such as combat robots, require their own unique size assortments. For example, you may need a set of 5/16 x 2" alloy steel (for strength) bolts and nuts to fasten the body to the frame. These should be purchased on an as-needed basis.

Fastener Materials

The most common metal fastener is steel, plated with zinc. These resist rust and are quite affordable, even in small quantities. Yet, there are many other materials and platings for fasteners, including aluminum, titanium, and even plastic. The following information applies to all fastener hardware: screws, nuts, washers, etc.

• **Zinc** plated steel is the fastener of choice because of its ready availability and low cost. Typical steel fasteners are made with 1006-1038 carbon steel, which exhibits high tensile strength, yet can be readily machined. Steel fasteners are magnetic, which can affect certain compass sensors in robots.

• **Stainless steel** offers resistance against rusting or corrosion. Stainless steel fasteners do not need to be plated because the material already resists rust and other forms of corrosion. Stainless steel fasteners are commonly available in any of several alloys: 316 stainless steel is non-magnetic and has extended corrosion resistance, while 18-8 stainless steel can be mildly magnetic, exhibits good corrosion resistance, and is competitively priced.

• **Brass** is a softer metal that's most often used for its appearance. No plating is necessary, as brass resists rust. Brass fasteners are naturally non-magnetic. Note that some brass fasteners are really brass-plated steel and some "steel" fasteners are really zinc-plated brass. Always look closely on the label to determine what you're getting.

• **Aluminum** is used when a metal fastener is desired and weight is a major consideration. It's also the preferred fastener for aluminum structures, as it will not cause corrosion, which can happen if you use steel fasteners with aluminum framing. Threads in aluminum fasteners are more prone to stripping, so avoid using aluminum in high stress applications.

• **Titanium** offers supreme strength for its weight. Titanium is only moderately heavier than aluminum, but has the strength of steel. Unlike those made from steel, titanium fasteners are not magnetic; however, some titanium fasteners are only titanium-plated steel. Also, steel fasteners can be tempered and hardened to improve their strength. As you can imagine, titanium fasteners can be frightfully expensive, but, when weight is critical, they're one of the best choices.

• **Nylon** (more generically called polyamide plastic) is considerably lighter than steel or brass and is advantageous when weight is a concern. Generally, nylon fasteners are available in two colors: natural (off-white) and black. Natural nylon is far more common and tends to be cheaper.

Obviously non-magnetic, nylon fasteners can also withstand higher temperatures than most other plastic fasteners — up to 300°. One use of nylon fasteners, other than lowering the weight of a robot, is as a low-friction glide.

• **Polycarbonate** is a high-impact thermoformable plastic. Polycarbonate fasteners are not common and they tend to be expensive. They are the best choice when high electrical insulation is required.

Sources

Aaron's General Store
www.aaronsgeneralstore.com
This is a "portal" to a number of online specialty fastener stores. The machine screws site sells Phillips, slotted, hex, Torx, and Pozidriv, in addition to SEMS screws, bolts, SAE, and USS washers and nuts, steel, and stainless steel industrial fasteners. Hard to find items are a specialty.

Allmetric Fasteners, Inc.
www.allmetric.com
This is the site for fasteners, metric style.

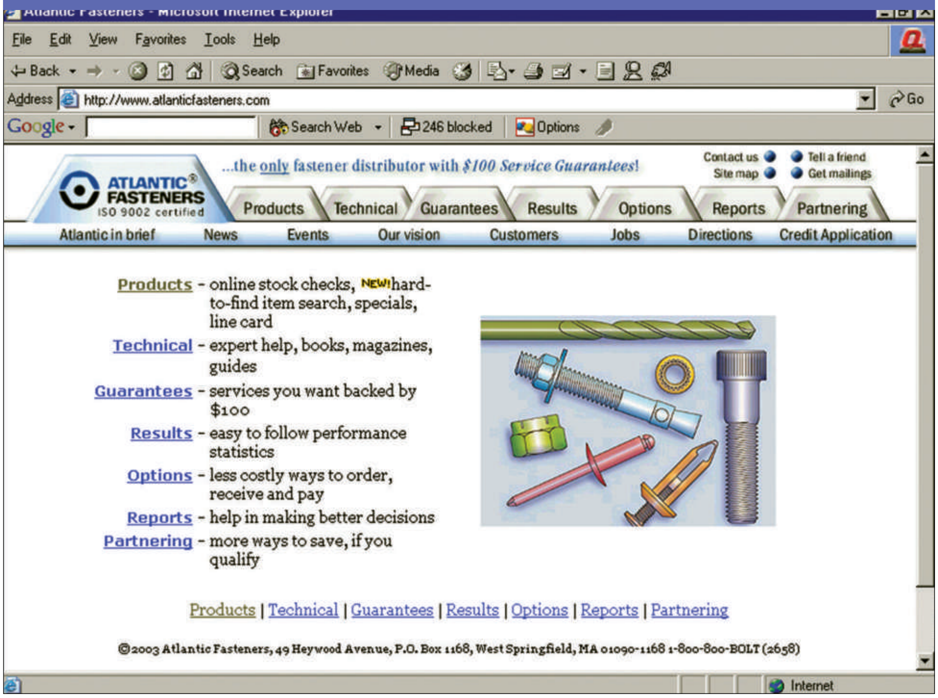
American Bolt and Screw Manufacturing Corporation
www.absfasteners.com
Suppliers of fasteners (bolts, screws, nuts, washers — you name it) to industry, they cater to high quantity purchases.

Atlantic Fasteners
www.atlanticfasteners.com
This site offers fasteners in tens of thousands of varieties, with pictures. A downloadable catalog in Adobe Acrobat PDF is available.

Barnhill Bolt Co., Inc.
www.barnhillbolt.com
They offer fasteners for all occasions, including all-thread, rings, threaded couplers, thumb screws, roll pins, retailers, and the usual nuts, bolts, and washers in zinc, steel, stainless, brass, nylon — both metric and standard.

Bolt Depot
www.boltdepot.com
Wood screws, sheet metal screws, machine screws, hex bolts, carriage bolts, lag bolts, socket head cap

FIGURE 1. AtlanticFasteners.com offers both standard and specialty fasteners. There are no online sales (ask for quotation), but they offer a method of browsing, with pictures.



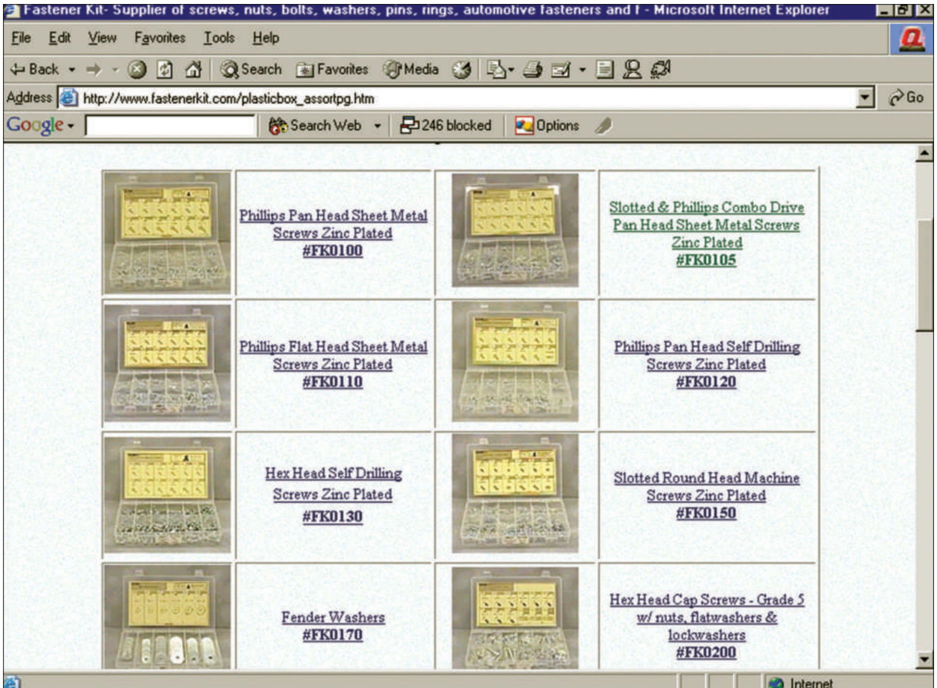


FIGURE 2. If you want several sizes of fasteners, **Fastenerkit.com** provides a variety of pre-made assortments.

screws, nuts, washers — standard and metric sizes are for sale by individual piece or in small quantity boxes.

Fastenal Company
www.fastenal.com
Fasteners, as well as industrial components and parts (casters, etc.), are available online, in addition to the local outlets in many US states.

Fastener-Express
www.fastener-express.com
Fastener assortments, socket screws, metric fasteners, aluminum fasteners, servo and flange screws, machine screws, sheet metal screws, nuts, washers, and nylon fasteners are offered.

Fastenerkit.com
www.fastenerkit.com
Fastener kits, bolts, nuts, washers, and clips are here.

Fuller Metric Parts
www.fullermetric.com
While looking for metric fasteners of all sizes and styles — including pins, threaded spacers, and socket head screws — check out their tech info pages.

Maryland Metrics
www.mdmetric.com
Something of a one-stop-shop, Maryland Metrics carries bearings, linear bearings, fasteners, rods, gears, pneumatics, and more.

McFeely's Square Drive Screws
www.mcfeelys.com
Check out the technical information about screws while looking for fasteners, tools, adhesives.

Micro Fasteners
www.microfasteners.com
Fasteners (machine screws, nuts, lock

washers, rivets, etc.), predominately in petite sizes, are available in US and metric threads.

Micro Plastics, Inc.
www.microplastics.com
A major manufacturer and seller of plastic fasteners — including clips, cable ties, hose clamps, plastic stand-offs, panel fasteners, whole plugs, threaded rod, and the usual screws, nuts, and washers — they have products available in standard and metric sizes.

MSC Fasteners
www.mscfasteners.com
Fasteners offered here include: body washers, button heads, socket cap screws, lag screws, carriage bolts, levis pins, cotter pins, drive screws, flat head socket cap screws, hex head cap bolts, and more.

Wicks Aircraft Supply
www.wicksaircraft.com
This is a small aircraft (not hobby) parts and specialty fasteners supplier.

Small Parts, Inc.
www.smallparts.com
This site has a variety of components, including a large inventory and variety of fasteners. **SV**

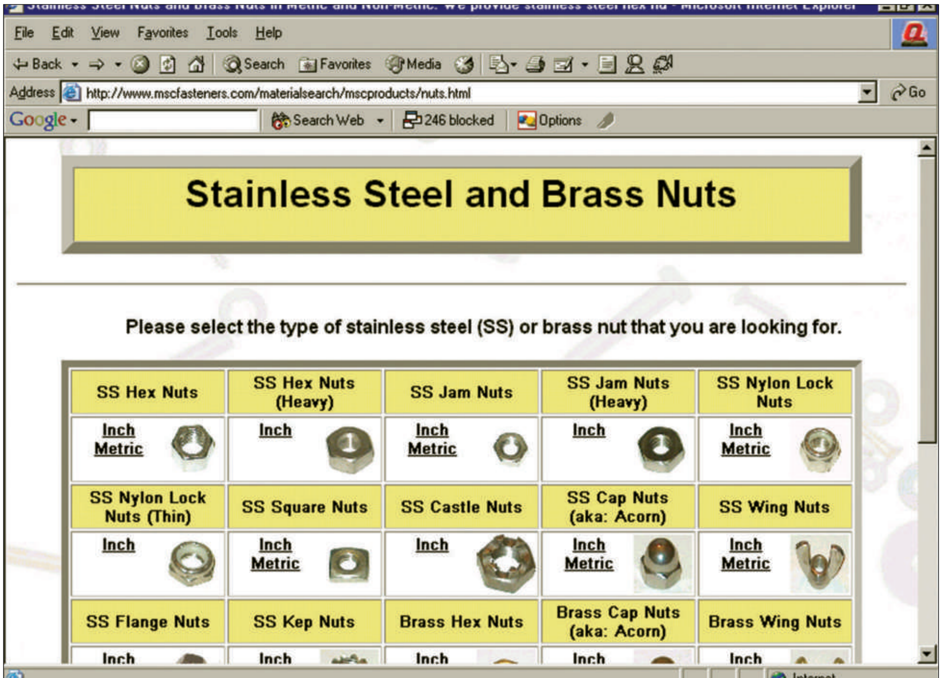


FIGURE 3. The picture-driven website at MSC Fasteners makes finding what you're looking for easy.

ABOUT THE AUTHOR

Gordon McComb is the author of the best-selling books *Robot Builder's Bonanza*, *Robot Builder's Sourcebook*, and *Constructing Robot Bases*, all from Tab/McGraw-Hill. In addition to writing books, he operates a small manufacturing company dedicated to low-cost amateur robotics, **www.budgetrobotics.com**
Contact him at robots@robotoid.com

Robytes

by Dave Calkins

Another month, another collection of exotic robot trivia. But this stuff isn't always easy to come by. Got a good story on robots? Email me: news@robotics-society.org If you'd like to get even more robot news delivered to your in-box (no spam, just robo-news), drop a line to: subscribe@robotics-society.org

— David Calkins

The Tin-Can Can-Can?

Well, this article has shown you robots that can walk, run, crawl, and even surf. So, dancing robots seem like the next logical step. Parallax has granted your wish! Most of you know Parallax from their BASIC chip sets (e.g. the BOE Bot). Who knew that they were also coming out with a precision dance team?

Ken Gracey has taken their Toddler line of walking robots and taught them a few tricks. "Rather than make a task-oriented set of robots with a specific project in mind, my idea was to make entertainment robots." You stick your left servo out, you stick your left servo in ...

Photo courtesy of Parallax, Inc.

Yes, they do the Hokey Pokey dance! A single blue robot leads and, using transmission and reception antennae, the backup black robots follow his moves in a synchronized dance style. Meanwhile, the lead robot sings the Hokey Pokey song in an appropriately annoying, high-pitched tone. You can watch a video of the dancing bots at **http://servo01.notlong.com** Obviously, this technology can be used for far more than dancing, but showing them doing a kids' dance makes it a lot more fun.

If they learn to do the Macarena, I'm outta here.

Speaking of Coordinated Robots ...

Photo courtesy of SRI, Inc.

SRI International has teamed up with Stanford, the University of Washington, and ActivMedia to create the Centibots! The general goal is to advance SRI's knowledge of distributed robotics. Centibots are a set of 100 autonomous robots designed to demonstrate coordinated mapping, tracking, and guarding in a coherent fashion. Missions for the bots include search and rescue, as well as surveillance. ("erm ... Guys? Is it me or do you get the feeling those robots aren't just vacuuming ... ") The robots are broken up into two groups — a smaller group of 20, which initially scans the area and collectively builds a map of it, and a second, larger group of 80, which searches for things, discovers intruders (or occupants, I suppose), and reports back to the command center.

If one bot gets destroyed or

breaks down, the system has its tasks farmed out to other bots. The robots can be sent into areas which are otherwise unsafe for humans (collapsed or earthquake-damaged buildings, chemical-spill sites, burning buildings, terrorist-occupied structures, my editor's bathroom ...) or areas where people would not be able to see anything (such as smoke-filled buildings). The bots, using advanced sensors, could see through the dust or smoke.

I just hope my editor doesn't get a set of them to figure out why I keep missing deadlines ...

Robot Intelligence "Rises"

How do you get robots to be smarter? Yeast, of course! (It rises ... get it? Never mind. I won't quit my day job.) Seriously: Scientists at the University of Manchester have created a "robot scientist" that can creatively think — in this case, generating hypotheses about genes contained in baker's yeast.

The robot then carries out experiments to test and confirm the hypothesis. "This research is very exciting, as we have handed control of the experiment over to the robot — so there is no human intellectual input in the design of the experiments or the interpretation of data," says Professor Ross King.

Based on pre-programmed knowledge of biochemistry, the robot comes up with several hypotheses and then eliminates the inaccurate ones via experimentation — saving time and money. This is much like the way that scientists would conduct similar experiments, only the robot can work all night (just like grad students, only without the pizza).

In the long term, such robots could conduct far more experiments in much less time and develop much more knowledge about baker's yeast and many other things (Such as, say ... brewer's yeast?).

And, of course, eventually they'll put everyone out of a job, except me (No robot could ever make jokes as bad as mine!).

Dean Kamen Finally Finds a Market

Photo courtesy of Segway, LLC.

So far, only 6,000 Segways have found their way into the hands of civilians, but the military is about to buy a whole bunch more.

Segway, in conjunction with DARPA and Carnegie Mellon, is developing "Segway Soldiers".

Engineers have added laptop computers to control the platforms and loaded them up with cameras, sensors, communications gear, and other gadgets — turning them into rolling combatants.

Much like many other forays into military robotics, researchers hope to use the seg-bots to perform dangerous battlefield missions.

They've already programmed them to avoid obstacles, track and chase things, and even open doors by themselves.

Of course, they don't have to be offensive — researchers also envision them helping injured humans off the battlefield.

For more information on Segways in military applications, read Edward Driscoll's article in the February 2004 issue of *SERVO*. SV

Has The Father of Robotics Found Its Killer App?

by Edward B. Driscoll, Jr.

Looking for the first "killer app" for robots? Waiting for robots to show up in homes across America? Dr. Joseph F. Engelberger thinks he has the answer — and, if he has his way, a robot, or a few of them will be in your home by the end of the decade.

During a recent interview, Engelberger described himself as, "the nominal 'father of robotics,'" which is certainly a reasonable title, considering that his first company, Unimation, pioneered the installation of robots, called Unimates, on an assembly line for General Motors in 1961. While factory robots have grown to an 8.5 billion dollar industry, encompassing an estimated 200,000-plus robots now in service, Engelberger believes that the most pressing need for robots is in assisting the elderly and infirm. Such a class of robots, developed as projected, could dwarf the number of robots in factories. To this end, Engelberger's second robotics company has already made inroads into hospitals, via his Pyxis HelpMate® robot from Cardinal Health (**www.pyxis.com/products/new helpmate.asp**).

If you've ever seen a PBS or Discovery Channel documentary about robots, you have seen squat, square, four-foot high robots transferring medicine or food from one section of a hospital to another; these are Engelberger's robots in action. These hospital robots were built by HelpMate Robotics, Inc., which Engelberger founded, but has since sold to Cardinal Health.

The Pyxis HelpMate robot inspired Engelberger to create more versatile home healthcare robots.

Providing a New Lease on Life

Both of Engelberger's corporations were profitable enough to be acquired by larger firms. Today, Engelberger wants to start a third corporation with the purpose of putting a personal service robot in your home. Engelberger describes a robot that will support a homebound person, allowing that person to live independently longer than would otherwise be possible, at a much lower cost than a nursing home or human live-in help. Engelberger isn't looking to build the robot equivalent of a home healthcare aide.

His home healthcare robot, unlike the Pyxis HelpMate robot, would fetch objects — such as food or medicine — in addition to monitoring its infirm owner. "There are things like that which are relatively simple.

Dr. Joseph F. Engelberger — a.k.a. "Daddy", if you're a robot.

"You know, people say, 'You might not be able to do this or that,'" Engelberger explains. "Well, I don't know how to have a robot bathe a person — not now; the technology's just not good enough. It probably can't dress a person, either, but it could do enough that a homebound person could get away with about a couple of hours a day [from] a loving relative, or maybe someone from the visiting nurses association, and that's the goal."

How Much Will it Cost?

Engelberger believes that, "We can have a full-fledged, mobile, sensate, articulate, two-armed robot commercially available at an attractive price." He envisions these robots at a cost which an insurance company or Medicare will cover for the average person. Engelberger is planning for his robots to have a lease price of \$500.00 to \$600.00 per month — about the same as a luxury automobile lease and much less than an extra four to eight hours a day of conventional home care. Along similar lines, Engelberger estimates that gearing up to build his robot would cost less than gearing up for an assembly line in Detroit. "It wouldn't cost as much as building a Cadillac. It's smaller and the only thing that's different is that it's got a helluva lot more software — and software, you only do once."

The Market is Vast

Of course, it's on those assembly lines where robots have made their greatest inroads. It has only been with recent products — such as the Sony Aibo robot dog and the robotic Roomba vacuum cleaner — that mobile robots have gotten a toehold in the home; while those products will eventually be looked upon as important in the evolution of commercial robots, they're also being looked at the same way as a mid-70s TRS-80 is compared to today's tricked-out PCs with 4 GHz Pentium processors. Engelberger believes that is ripe for change. "Sure, there will be people who want a robot around the house to mix drinks for their guests," but the big market for robots is in service to an aging population. "[T]he market is vast, if you consider that the fastest growing age in the United States is 85 and most 85-year-old people have some kind of handicap."

It's to this end that Engelberger recently allowed his name to be used on the trophy prize in the DIY Network's new series, *Robot Rivals*. He sees the show, which features teams of college students building robots to solve problems, as doing a great deal to promote the practical applications of robots. "The robotics industry is going to continue to need new participants — bright young people — to join the fray. Contrary to many areas of scientific endeavor, there just is no plateau that we're going to have to get to. I'm sitting here, pointing at myself, and I say that the perfect robot is one of these — a human being — and that's a hard job to re-create! But, we'll get better and better at it and I think that programs like this are going to help fill the pipeline with students who will deliver the line of necessary hardware."

The Tip of the Iceberg

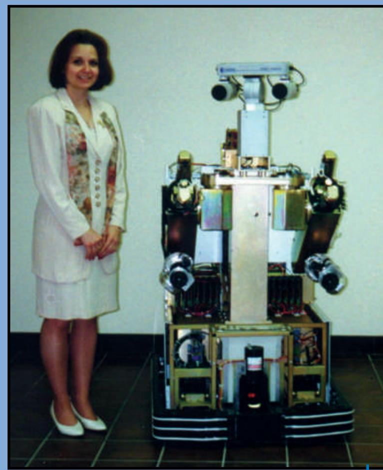
The recent glut of Robot TV shows on cable and the introduction of the Aibo and the Roomba are merely the tip of the robotics iceberg. Evolution Robotics, Inc., of Pasadena, CA (**www.evolution.com**), seeks to revamp robotics development by eliminating the redundancies of programming operating systems (see the August 2002 *The Nuts & Volts of Amateur Robotics supplement*). Their mantra is that robots are what computers were in the late 70s — a hobbyist's market, eager for a standardized platform that would put the emphasis on new applications, rather than simply trying to get a device built from scratch. While it's far too soon to tell if their strategy will be successful, the key is the creation of a robot operating system that they hope will be the equivalent of what Windows is to personal computers: a standardized system that applications can be built on, rather than reinventing the wheel for each new robot design.

Here, the Pyxis Helpmate is in action, aiding in serving hospital meals. Maybe 'bots will know what's in that meatloaf ...

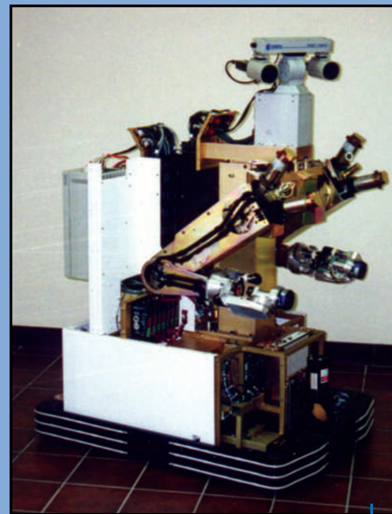
Rollin', Rollin', Rollin'

Software truly is the key to a successful robot, according to Engelberger, who feels that, while research into such areas as two-legged mobility and facial expression is useful, these fields may be dead-ends for building practical home robots or, at the very least, slow their development and acceptance. Engelberger uses the example of the millions of dollars that Honda has poured into the development of Asimo, a human-shaped walking robot (**world.honda.com/ASIMO**). "I don't want to see a two-legged robot," Engelberger says. "I feel very strongly against legs. Now, that sounds odd, especially when so many Japanese researchers are making legged robots, but everything is wrong. See, the legged robot has all [of] its battery power on its shoulders, which makes it very unstable. A wheeled robot has all its battery power well below its ass, down near its wheels and it becomes very stable." One aspect of anthropomorphizing is important for tasks: arms. "You need two arms for a number of tasks. You need

The Pyxis Helpmate also assists in medication distribution, reducing human error and overload.



Engelberger's new vision of a home healthcare robot is more adaptable to the needs of the patient.



Remember breakdancing? Engelberger designed these 'bots to be stable, mobile, and prehensile.

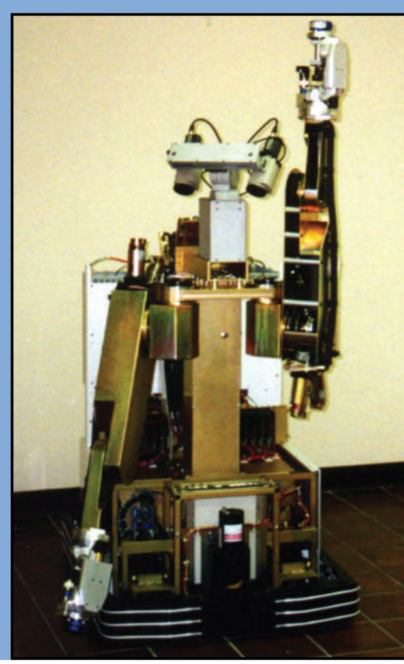
sensor perception — visual, tactile, and auditory. So, it's two-armed, it's mobile — but probably with wheels — and sensate, and articulate." It will also have voice recognition, if Engelberger has his way. Speaker-specific voice recognition is quite difficult today, because the software can't recognize a myriad of differing accents and inflections. "But, if you were a personal servant to two or three people in the home, you would know all their accents, you'd know all their voiceprints, and you'd be able to accept commands that are quite sophisticated." Will people accept a robot with the aesthetics of a rolling refrigerator in their homes?

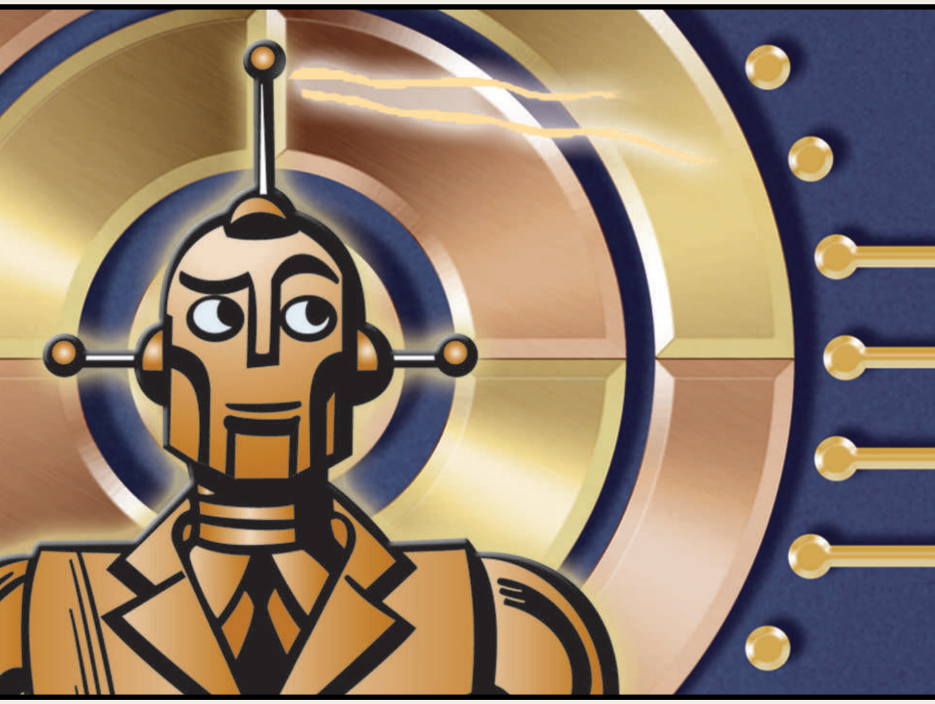
Regarding his hospital robots, Engelberger says, "We put stripes on it, like a volunteer nurse — running around the hospital, going up and down on the elevators, and going to all the departments. It's just amazing how children and even grownups will anthropomorphize it, because it speaks, and because it moves and it is non-threatening." You can see a similar trend in the anthropomorphizing of non-human-appearing robots in the movies, as well.

As stated in the premiere issue of *SERVO*, the squat, fire-plug-shaped R2-D2, who speaks in nothing but beeps and boops, is a far more beloved character than C3-PO, who speaks English and has a much more human appearance. Like the robots in *Star Wars*, robots back on planet Earth will eventually have many different appearances — and functions.

Some of those robots could, indeed, be rolling into your home in the not-too-distant future. If robots are in the same position as personal computers in the 1970s, it means there's a lot of room for growth in the next decade or two. **SV**

Raise your hand when we call your name!





Humanoid Robotics Design Considerations

by Karl Williams

MAKING THE CASE FOR ANTHROPOMORPHIC ROBOTS

Is it possible to create an artificial human — an android? The human body, including the brain, is one of the most sophisticated biological machines in existence. The design of the human body is so complex that, to build comparable machines, our technology would need to advance significantly. With our current understanding and technology, we can only attempt to mimic the form of the human body, since it is impossible to recreate it at this time. It is no wonder that man looks to nature for insight and inspiration when designing machines.

The idea of "reverse engineering" humans has fascinated mankind for a long time. The concept of how the human body functions has proven to be extremely complex. For example, the human hand and wrist are very complicated devices for grasping and moving objects, but, when implemented in machinery, they must be simplified in order to keep the mechanics and control systems within a reasonable level of complexity.

There are many reasons for wanting to create humanoid robots. Building robots that have a human form would allow those machines to take advantage of all the tools and equipment that have already been developed for humans. One of the main motivations for creating androids is the psychological aspect of human interaction with machines. We are much more comfortable communicating with machines that more closely resemble the human form, as opposed to machines that have an almost alien and sometimes frightening appearance.

How often have you heard the phrase, "It's almost human," when watching a robot do something interesting on a television show or at a science center? We humans quite often project our humanity onto machines and other life forms that resemble us. Another phrase that is heard quite often is, "It has a mind of its own," when watching an automaton perform some entertaining task or a robot that senses and responds to its environment.

People would be much more comfortable interacting with machines that are designed to look like the human form. Now that automated banking machines have eliminated most of the human tellers, wouldn't it be nice to be able to deal with banking machines in a face-to-face manner and input your data without having to fumble with a card, cramped keyboard, and a small monitor? Imagine being able to walk up to a humanoid robot, have it access your banking information via facial recognition software, and then verify your identity with a retina scan. You would be able to talk to the machine in exactly the same way that you would with a live person.

In 1942, Isaac Asimov published his three laws of robotics in a short story, called "Runaround," which was published by Street and Smith Publications. The three laws were stated as follows:

1. A robot may not injure a human being or, through inaction, allow a

human being to come to harm.

2. A robot must obey the orders given to it by human beings, except where such orders would conflict with the first law.

3. A robot must protect its own existence, as long as such protection does not conflict with the first or second laws.

These laws could be incorporated into a set of rules defining robot morality, that is, if the robots being built are intended to respect human life.

Most roboticists no longer agree with Asimov's laws. The first law disqualifies several important roles that humanoids are well suited to perform, such as soldier, police officer, or security guard. Much of the government funding for robotics, provided by the Defense Advanced Research Project Agency, is focused on military applications. The cruise missile is the perfect example of a fully autonomous robot that follows the first part of law two, but, in doing so, breaks laws one and three. The Predator robot, developed by the Central Intelligence Agency, is another example of a robot that kills with deadly precision by launching Hellfire missiles at its targets. It could be argued that these kinds of robots are now a necessity in the war against terrorism and rogue military nations which threaten national security. To build an artificial person or humanoid, we must first consider what it is that we are trying to construct. To answer that question, we need look no further than ourselves. The requirement specification would look somewhat like the following list:

1. ***The robot should have a more or less human form.*** It should have two legs, two arms, a torso, a human face, and a head. It should be roughly between 4 and 10 feet tall. The overall look of the robot should not stray considerably from what would be considered acceptable human appearance.

This is important, since some people are afraid of robots that remind them of creatures like spiders, snakes, and lizards. (*I'll take Aibo over a 10-foot tall humanoid any day, thanks — Ed.*) I would consider one of the main motivations for creating humanoids to be the psychological aspect of acceptance of the machines by humans.

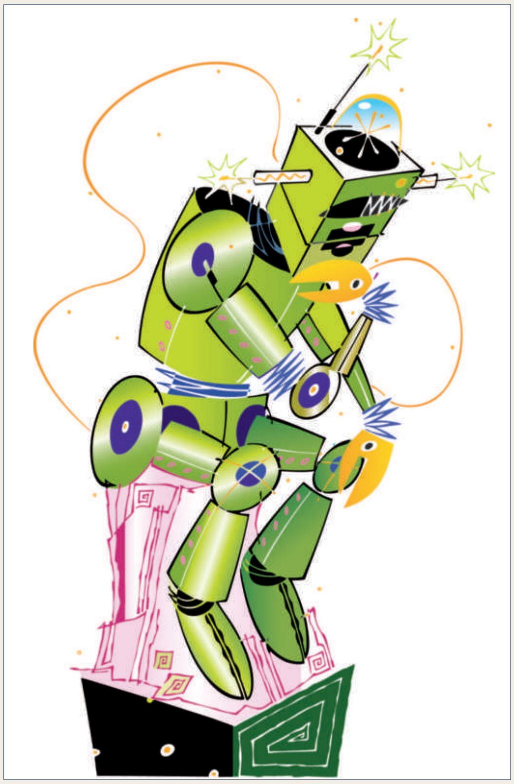
2. ***It should be able to communicate with humans in their native languages, without the use of an input device***, like a keyboard. The robot should at least respond to spoken commands and it must be able to generate language by speech synthesis of some sort. It should also be able to convey simple emotions through facial expressions which correspond to the generated speech.

3. ***It must be able to move from one location to another, under its own free will or at the command of a human.*** While doing this, it must not harm any other objects or step on humans or pets.

4. ***It must be able to sense its environment and avoid obstacles and dangers which it may encounter along the way.*** A flight of stairs is not a problem for most healthy adults, but could pose a catastrophic encounter for a humanoid robot.

5. ***It must be able to pick up and carry objects in order to do some useful work***, such as vacuuming the carpet or cleaning the toilet. Robotic arms and hands will be necessary to accomplish these tasks.

Is morality as vital as intellect to the evolution of robotics?



6. ***The humanoid should be able to learn from its own experience and retain that information.*** It could then conceive its own strategies for dealing with those situations in the future.

7. ***The robot must possess some manner of intelligence and the flexibility to accept training, while adapting to the tasks that we wish it to perform.*** This would include solving simple problems that it might come across while carrying out its tasks.

8. ***The humanoid should understand and obey the basic principles of human social interaction.*** It should follow an acceptable code of behavior and possess a set of morals.

The attributes listed above are a very basic set of requirements. There are many other components that would need to be included to create a machine which would even come close to having the capabilities of a human.

There are many issues to consid-

The first step in humanoid robotics design is the development of effective bipedal movement.

er when setting out to design and build a humanoid robot. Some of the considerations are more philosophical in nature than the mechanical, electronic, and programming achievements needed.

Whenever I would talk about building humanoid robots to my friends, most of them would comment that it would be nice to finally have a robot that could do some work around the house. They talked about a machine that could wash the dishes, take out the trash, vacuum the floor, prepare the meals, cut the grass, and tend the bar, etc. When you mention humanoid robots, that is the kind of machine most people think of. My friends would also point out the amazing abilities of Honda's humanoid, Asimo. Why aren't these machines available and working in our homes? I would explain that it would cost millions of dollars and take considerable time to build something comparable to the Honda humanoids.

The fact is that the technologies necessary to build smaller scale humanoid robot projects have become relatively inexpensive. With the right amount of imagination and innovation, anyone can create amazing machines in their basement laboratory.

The robots of the future are now within our reach, because we can build them ourselves! Robotics is a unique area of study because it encompasses many different disciplines, such as electronics, computer science, mechanical design, control systems, programming, and biology. This is what makes building robots so interesting and fun. The humanoid robot fascinates man, because it is a machine that closely resembles life and man himself. SV

GEERHEAD

by David Geer geercom@alltel.net

Any Bot He Can Build, She Can Build Better

Can gender engender a better-built bot? Let's find out!

Thank heaven (for lady roboticists). Yes, guys, women have innate interests in things like robotics and engineering. Heidi Schubert, our first example of a female roboticist, built her fledgling bot as part of an after school program.

And, no, we ain't talkin' *School House Rock* or *The After School Children's Special*, neither. Heidi — just your average aerospace engineer and Stanford Ph.D. holder — was first exposed to robotics in high school.

In an after school robotics class and competition, her challenge — and she did choose to accept it — was to construct a robot that could transport eggs (presumably, without breaking them).

Her First Bot

As robot intellects go, this first creation had no intelligence of its own. It appears that this was the perfect bot IQ for the task at hand.

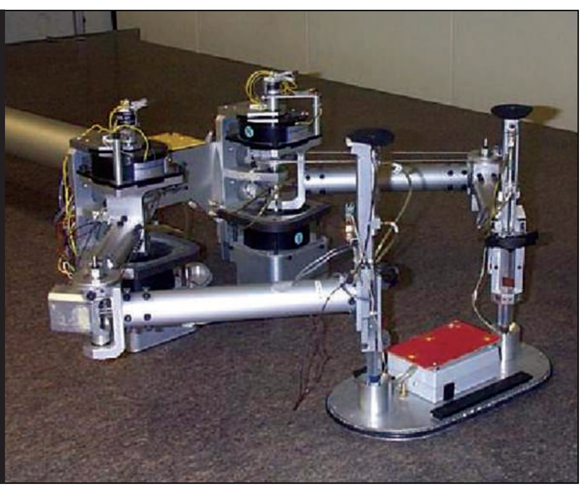
Rather plastic in personality, this "egg-xcursion-ary" bot was made of hydraulic syringes and plastic tubing for manipulation and mobility, as well as from other junk parts. This little bugger worked its egg-carrying heart out, without a whimper or a "cr-r-r-r-ack!"

Stepping Up at Stanford's Aerospace Robotics Lab

At Stanford, Dr. Schubert jumped in with both feet. She took on a Ph.D. robotics project that would cover all the pertinent areas of engineering, in addition to aerospace robotics.

In fact, it was robotics that gave her the opportunity to cover so much of engineering in a single area. The result of Heidi's work in hardware design, software design, and sensing control on this project was the Macro/Micro Manipulator.

This PhD project was solutions-oriented. The problem — space robots need to be big and bad, light and dexterous, all at the same time. For example, the Space Shuttle robot is large and light, but not dexterous at its endpoint.



MACRO/MICRO

Apparently, we've been solving that problem by making some astronaut be that dexterous endpoint. Considerate folk that we are, we'd rather not continue this.

So ...

The arm that will be used on the Space Station will have its own endpoint dexterity — human guinea pig not required. A micro robot known as the Space Station Remote Manipulator System (SSRMS), with a micron known as the Special Purpose Dexterous Manipulator (SPDM), makes up this arm.

Heidi's purpose in her project was to study how macro and micro robots can work together. On this understanding, her Macro/Micro was born.

Philosophy and Construction

To mimic the dynamics of a robot in space, the robot was designed to be a two dimensional creature — without vertical movement. In this way, the vertical effects of a gravity-free environment were removed from the equation.

To duplicate the horizontal responses the robot might have in space, its primary arm floats above a table. If the motors are off and you impact the robot, it floats away from you, just as it would in space.

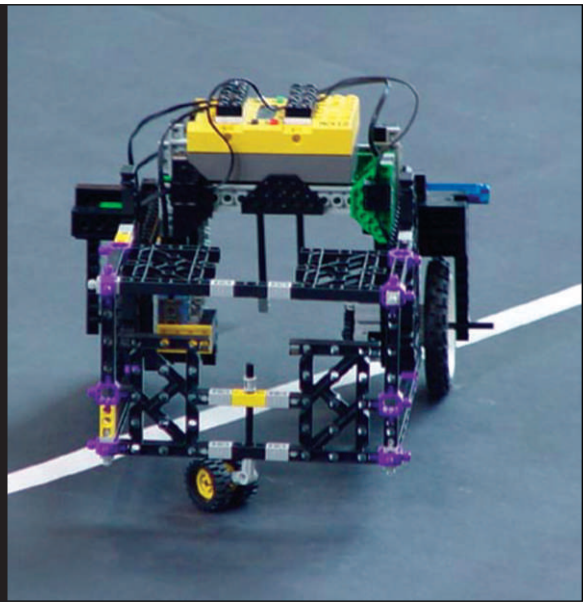
The robot uses plungers at the end of a pair of one-foot arms, which bend at their halfway points. With these, it can grab and move a floating object, which mimics a satellite.

The all-aluminum arm has other features ours don't — eyes. Each joint has a sensor. There is a camera over the robot and sensors, which rest on each smaller arm that extends from it. All these sensors work together to provide detailed information for precise movement.

Never Neglect the Value of a Big, Red Button

Like a big, red stop sign, the big, red button is there to stop the robot, should things go haywire. Of course, you have to have your hand on the big, red button or you might experience something like the following:

Once, when working on the robot's autonomous control, Heidi didn't have her hand on the red stop button and this was the time when she needed it. She had to chase the robot around — or should I say, chase the red button, or both — to shut the robotic arm down.



SIX PACK

This Isn't Kid Stuff

Sheila holds a BS in Aerospace Engineering. (What is it with women, space and robots? I think I smell a trend here. Hmmm?!) Any-way, in college, Sheila most enjoyed studies in orbital mechanics and jet engines (Don't all women?). Thanks to kids and Lego Mind-storms, however, today Sheila Doyle is hooked on robotics. Having bought the Mindstorms for her son, who didn't quite take to them, Sheila began working with the Legos.

Sheila claims not to be very up on tech smarts when it comes to building robots. She uses a Windows 95 box (that's tech for 'puter) for robot programming. Sheila inherited a laptop from her mom, complete with keys which have the letters worn off. She is also glad to have taken typing in high school.

Sheila started out in the Dallas Personal Robotics Group (DPRG) with her first competition in April 2003. Soon after, she went to a DPRG get-together — which, I'm sure, felt kind of like showing up at *Cheers*, as, reportedly, "Everybody knew her name!" With so few female members, Sheila definitely stood out.

A Parts Supply, She is Not

When Sheila finishes a new bot, she takes it to her younger son's school to demonstrate it; it's part of something on the order of a Show and Tell. After one such demo, one of the kids asked her if she could bring a ton of robot parts to his birthday party the very next day. He wanted them so that he and his little party guests could build a bunch of robots together. Can you guess her reaction?

Work and Play

In her off time, Sheila, a power plant engineer by trade, builds robots just as she sees them in her head. Some examples include line and wall followers.

Other bots include a toilet-paper-tube-picker-upper, which doubled as an in-the-flour-can-toilet-paper-tube-dropper (not where I keep mine, but hey). This bot used a laser pointer to find aluminum foil targets on the tube and can.

Challenges in building this bot included constructing the part that would grip and move the tube securely and enabling the bot to operate precisely.

One of Sheila's other bots locates soda pop cans and is able to carry an entire six pack. It avoids walls and locates cans using sonar. This bot, appropriately named Six Pack, follows a preset course and avoids walls using its sonar capabilities.

Six Pack has competed in the DPRG's Can-Can competition at RoboRama 2003.

Sheila is currently hard at play on a bot that will pour glasses of tomato juice. She is also building sensors for a line following bot.

HEIDI SCHUBERT

Aerospace Engineer and Stanford PhD Holder.

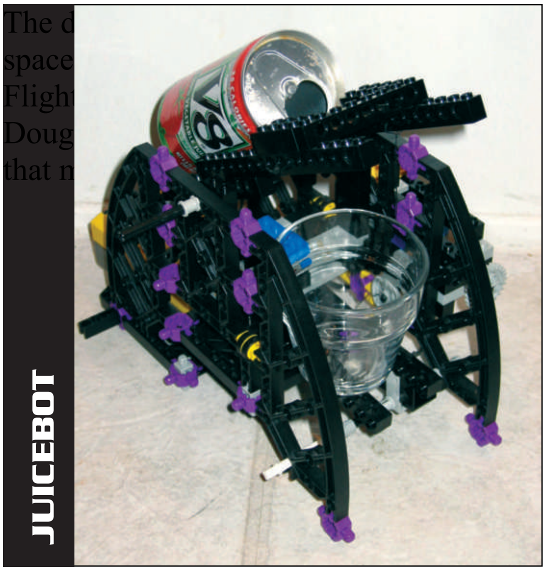
First Bot

A robot that transports eggs without breaking them.

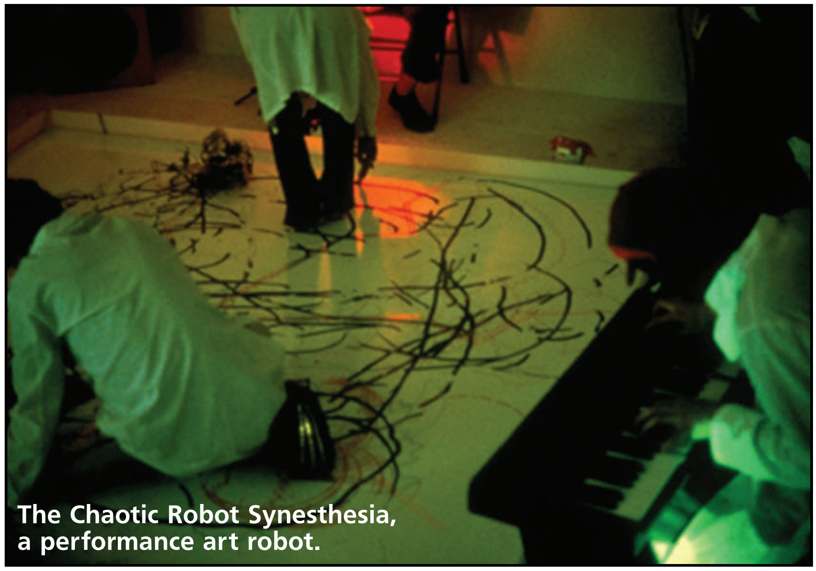
Highlights

Heidi is working on a micro robot known as the Space Station Remote Manipulator System.

heidi.schubert@stanfordalumni.org



JUICEBOT



The Chaotic Robot Synesthesia, a performance art robot.

Nicole and Leesa Abahuni — Art-Boticists??

Artists, roboticists, and twins, Leesa and Nicole Abahuni were born in Long Island. Their art is researching the senses in a broad array of contexts, using electronic performances that star their own robotic creations. The Abahuni sisters, also known as The Turbo Twins, have exhibited their performing bots at shows ranging from the Half Machine Festival in Copenhagen to the 28th Annual Siggraph Digital Arts Festival and Convention in Los Angeles, CA — and many more. Both ladies are graduates of the School of Visual Arts, NYC, with Bachelor of Fine Arts Degrees (BFAs) in Computer Arts. The Turbo Twins — a moniker bestowed by technicians in the Sculpture Department at their Alma Mater visual arts school — have a reputation for constantly running around with power tools, creating robots and other strange mechanisms of metal, plastics, and electronics.

Influences

You know you always have to ask artists about their influences, right? The Abahunis were inspired by what they saw in an art history course — the mechanical sculpture of Jean Tinguely. Named "Homage to New York," the self-destructing masterpiece tore itself up and equally affected everyone fortunate enough to witness its performance first hand.

Bots and Builders, Just Working Together

The Turbo Twins and their creations work together to form works of art. Their robots have enabled them to develop a broad artistic ensemble of tools with which to share ideas about combining humanity and technology.

The Abahunis have traveled the world, showing their robots. Their work has even introduced them to Sultans, who have enjoyed their performances. They are eager to meet and work with other roboticists who combine art and technology.

Photo courtesy of Jacob Tendy

SHEILA DOYLE

Power Plant Engineer, Mother, and Bot Builder.

One of Her Many Bots

Six Pack locates soda pop cans.

Highlights

Some of her bots include line and wall followers. Another is a toilet-paper-tube-picker-upper.

srdoyle@earthlink.net

Introducing Their Bots

Once called a scratch-and-sniff robot by one of its fans, Linus is a highly perceptive sensing robot. Linus is constructed of scraps of plexi and ragged wires. The wires enable the addition and subtraction of sensors, appendages, and parts.

Minus is a minimal (-ist?) robot, not good at following directions. Mounted on a clear,

Re-capacitance, performing in the United Arab Emirates.

LINUS

frosted plexi base, Minus' gears make loud snorting sounds when moving. It frequently does spirals. I would have to guess that Minus' personality is the robotic equivalent of a camel's or an ostrich's.

Primavera, another plexi and wire bot, bounces off the walls like a drunk looking for some way to go that does-n't hurt.

Ladybug rides on opaque red plexiglass. On one trip, she busted her gears trying to do a difficult reverse movement action with a weight imbalance. She convulsed around her drawings, which ended up looking like hand-drawn scribblings.

Victrola rests on a vinyl record and carries a wireless mic. This setup allowed The Turbo Twins to mix the sounds of the robot's gearbox with those of hired musicians during one of their performances.

Performance Art Robotics?

The Abahunis' performances have mixed robotics, music, color, artists, and musicians. In one performance, the robot in question moves over black drawings that artists are simultaneously creating on a white platform.

As the robot chooses which lines to follow, it marks over them with a marker. As the performance continues, this drawing moves over a diagram of seven notes, which are represented on the platform by a color wheel. The robot's springs complete circuits in the color wheel as they move over them, triggering a light of that color. The colors correspond with notes, which the musicians then play.

Listen to the Muse-ic

The Abahuni twins listen to AC/DC when building their robots. When rethinking their work, however, they put on their serious, white lab coats and listen to modern classics, like the works of Philip Glass.

Speaking of music, whistles may call dogs, but the gentle hum of certain robot motors can put them to sleep. The Abahunis share a dog, which was once hypnotically entranced by one of their bot's motors. The dog actually fell asleep right in front of the robot, which was spinning in circles.

Of Interest

The Abahunis' robots neither compete, nor do they perform precise

NICOLE & LEESA ABAHUNI with Billy Kluver
Artists, Roboticists, and Twins.

Some of Their Bots
Linus, Minus, Primavera, Ladybug, Victrola ...

Highlights
These sisters combine art and technology to create robots that perform.

nicoleesa@lycos.com

Lessons From the Ladies

The Moral of the Macro/Micro Story:
(Dr. Heidi Schubert's Robot Project)

Use dynamic modeling to its fullest to enable your robot to perform a variety of functions quickly and smoothly — and make sure to have lots of sensing.

The design of the Macro/Micro provides a great lesson for amateur roboticists. If you want to build dexterity into the functionality of your robots, study the Macro/Micro and the human arm. Consider using a large arm for large movements and attaching smaller arms or fingers to it for small, precise movements.

Sheila Shares:

Sheila Doyle recommends running your bots frequently, particularly the ones that use optics. Variances in background lighting will cause a variety of readings.

Sheila recommends the bot instructions available in Dave Baum's book, *The Definitive Guide to Lego Mindstorms*.

The Turbo Twins Philosophize on Bot-Making:

As if it wasn't obvious from their own creative flair, the girls recommend risk-taking, experimentation, and individualization. Sketch what you really want your robot to be, from head to ... whatever.

Ask yourself, what can your robot do that no others can? Does it have purpose? If not, perhaps you should give it one. Are you researching, examining, learning, and sharing something with robotics? What is unique about your builds?

operations.

Rather, it's what events happen by chance or which decisions the robot makes that take center stage. These robots are purposely flawed, as they are imbued with character. They cut against the grain, fulfilling the opposite of what roboticists are instructed to do. Their creators observe and enjoy the results.

These robots come with freaky long hair, in the form of

RESOURCES

Heidi's Page at the Stanford Aerospace Robotics Lab website:
http://arl.stanford.edu/projects/large_macro_mini_manipulator/superflex.html

Dave Baum's book, *Definitive Guide to Lego Mindstorms*, is published by APress.

Dallas Personal Robotics Group's Competitions page with great bot pics:
www.dprg.org/competitions/index.html

School of Visual Arts, NYC, MFA Computer Art Department:
www.sva.edu/mfacad

very practical — yet freely dangling — wires. These are bent into specific shapes, so that the twins can determine which parts are getting power and where the wires and the power are going.

Longer wires enable them to add and subtract arms and to rebuild the main body. In some cases, the extra arm is a magic marker used as a castor wheel. They also record the robot's thought processes as it moves around the floor. Elements of different kits and other part sources are built in, and out, and into the robots again for intentionally unpredictable results in behavior. Their five robots' personalities are shaped and reshaped, as if they were maturing with each new build. **SV**

The Influence of Billy Kluver, Engineer and Artist's Friend

It has been mentioned that the Abahunis were influenced by the work of Jean Tinguely, namely by her famous self-destructing, mechanical masterpiece, "Homage to New York". Tinguely, the Abahunis, and many other artists over the years have also been heavily influenced and aided by an engineer who gave much of his life to supporting their work. His name was Billy Kluver.

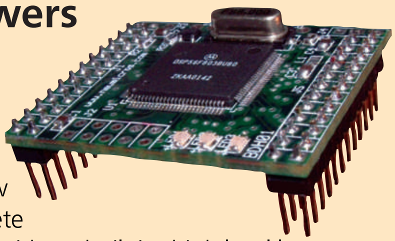
Mr. Kluver assisted Jean Tinguely with the engineering of "Homage to New York," which was performed in 1960 at the Museum of Modern Art in New York. A noted scientist at Bell Labs, Mr. Kluver was a pioneer of multimedia who worked with the artistic community to demonstrate to the world the powerful effects on the senses created by combining art and engineering — many of which used robotics.

Mr. Kluver passed on recently, leaving behind a grieving, yet grateful, community of artists, engineers, roboticists, media professionals, and family. Without his influence, much of the input for this article might not have been available, along with much of what we see and hear in multimedia today. These fields exist, in part, due to the nurturing presence of Billy Kluver.

New Products

CONTROLLERS & PROCESSORS

Plug-a-Pod™ — Plug-In Controller Powers User's Design



The new Plug-a-Pod™, a tiny 1.5" x 1.3" controller board from New Micros, Inc., is a complete microcontroller system with a built-in high-level-language and parallel processing operating system, IsoMax™.

Programmable in C, Small C, Forth, or Isomax, it can be used in a protoboard or stand alone, but is designed to be plugged into a user-designed carrier board for use as a controller module, thereby reducing total system cost.

The connections on Plug-a-Pod are positioned for easy routing on a two-layer board with the user's additional circuits. For instance, opto-isolation, communications drivers, high and low-side drivers, H-bridges, etc., and the needed connectors can be put on a two layer board and the complex, SMT, high density, thin trace, multi-layer requirements can be bypassed by adding connections.The board is 1.5" x 1.3". All connectors have .1" spacing. The pins plug into two dual row sockets, 0.2" x 1.2", with 24 pins each. It can also be soldered directly to the carrier board.

Hardware featured on one dual row set of pins includes: 16 General Purpose Digital I/O lines and one RS-232 serial channel. The first GPIO lines share functions with a four wire SPI Interface, six General Purpose Timers, and six Pulse Width Modulation (PWM) outputs. The additional dual row pins unique to the Plug-a-Pod include: eight-channel, 12-Bit A/D, and an additional eight GPIO. The PWM outputs are hardware based and can be used to control six R/C Servos or grouped to control three-phase Brushless DC motors, six PMDC motors, or complementary drive for the H-bridge for three PMDC motors.

The multimode timers can be used as three channels of Quadrature Decoders or Step and Direction counters. They can also measure pulse width, time ultrasonic ranging pulses, generate pulses, drive IR 40 KHz transmitters, etc. They can generate PWM outputs to drive another six R/C Servos. As the PWM and Timer modules are supported in set-and-forget hardware modules, the processor can perform higher level functions, such as acceleration-limited, velocity-profiled control of the moves of up to 12 R/C Servos with time left over for other tasks. Using the Timer modules as Quadrature decoder inputs, it can implement PID and acceleration-limited, velocity-profiled control of the moves of three axes of motion control at the same time, and other tasks, such as data collection on the A/D and individual GPIO management. The CANBus enables distributed processing networks, particularly in automotive and industrial applications, so parallel hardware and software can be combined.

IsoMax is real time programmable and based on state machine programming concepts. Development is interactive through the RS-232. It can interactively create new processor tasks — each being a state machine or thread — and then test that code. Virtually Parallel Machine Architecture (VPMA) is IsoMax's programming paradigm, allowing multiple background process machines to run independently in a virtually parallel fashion, handling tasks on the same level. IsoMax multitasks without a multitasking operating system; three lines of code equal a single state machine. The interactive foreground is always available for further development and checking.

Plug-a-Pod is thus very versatile and ideal for dedicated control of DC motors, BDCM, stepper motors, solenoids, and motion and control applications in general. The fast A/D works well for data collection and the CANBus enhances networked control applications.

The tiny Plug-a-Pod brings the computing and control function to a user's design. A single unit is $89.00 with linear regulators installed; other regulator options are available.

For further information, please contact:

New Micros, Inc.
1601 Chalk Hill Rd.
Dallas, TX 75212
Tel: **214 • 339 • 2204**
Email: **nmisales@newmicros.com**
Website: **www.newmicros.com**

Circle #117 on the Reader Service Card.

MOTORS

MAE™ Brand Hybrid Stepper Motors

All three series of new MAE™ brand hybrid stepper motors available from PennEngineering® Motion Technologies

can deliver accurate open loop control for high-performance positioning applications.

MAE "Y Series" motors (sizes 16, 17, and 22) incorporate AlNiCo magnets and are well-suited for high-speed operation; "N Series" motors (sizes 34 and 42) utilize neodymium magnets and exhibit excellent torque/rotor inertia characteristics for superior acceleration; "S Series" motors (sizes 22 and 34), with neodymium magnets, are optimized for maximum torque, as well as microstepping capability for smooth operation and fine positioning.

All MAE hybrid stepper motors feature step resolutions of 1.8º (200 steps per rev) in full-step mode, standard NEMA frame sizes, precision-honed stators, and ground rotors for tight air gap and maximum performance, a choice of 4, 6, or 8 leads upon request, single- or doubled-ended shaft extensions, and simple, rugged construction for high reliability and long service life.

Depending on type, these motors can range in length from 0.81" to 9.76" and achieve holding torques (bipolar) ranging from 10 oz-in to 3186 oz-in. Value-added customization options include gearboxes, encoders, shaft details, and leadwire-connector assemblies, among others.

For further information, please contact:

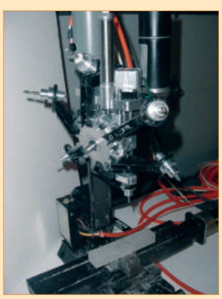
Penn Engineering Motion Technologies

343 Godshall Dr.
Harleysville, PA 19438
Tel: **877 • 748 • 8626**
or **215 • 256 • 6601**
Fax: **215 • 256 • 1338**
Website:
www.pennmotion.com

Circle #140 on the Reader Service Card.

TOOLS & TEST EQUIPMENT

Miniature Automatic Tool Changer



Home Shop Accessories, Inc. of Louisville, Kentucky has released plans and kits for the first fully automatic tool changer designed for miniature machining and CNC applications.

The tool changer is a complete, self-contained spindle and turret that was specifically designed to work with Sherline, Taig, MaxNC, and other mini CNC milling machines.

This six-station sequential tool changer will take your hobby CNC to the next level of performance and productivity. The mechanism can be used in full or semi-automatic mode and features a six-station tool turret and pneumatic drawbar.

The tool changer can function with CNC or manual machines and requires only one single actuation signal, either manual or remote, to execute a tool change cycle.

Highlights

- One channel of digital I/O (no subroutine programming or separate controller)
- Multiple kit levels are available for beginners through advanced
- Rapid tool change — one to three second cycle time
- Full or semi automatic operation
- Fast growing community of builders online at Yahoo minitoolchanger group
- Secure, fast online ordering with PayPal — receive your plans in a few minutes

If you would like to receive information on this product, send a self addressed, stamped envelope, marked Mini Tool Changer Info.

For further information, please contact:

Home Shop Accessories, Inc.

Mini Tool Changer Info.
4115 Dellridge Dr.
Louisville, KY 40207
Website:
www.homeshop accessories.com

Circle #126 on the Reader Service Card.

RoboBRiX Part 3

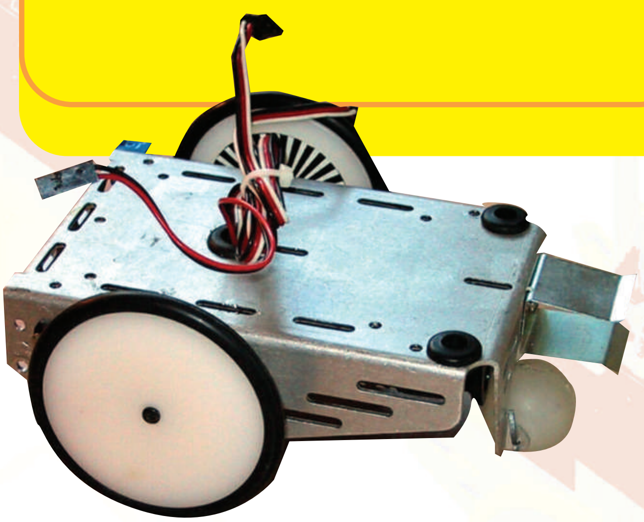
In this third article about RoboBRiX technology, we are going to build a wall-following robot.

by William Benson and Wayne Gramlich

Conceptually, wall-following is just a special case of line-following, but there are a few important differences that make this exercise interesting and worth the effort.

For one thing, you don't need to lay down a path on the floor for your robot to follow. Any old wall will do just fine and — unlike the line-follower that we discussed in last month's article — this robot will also avoid objects that are blocking its path. So, in both of these respects, this robot is more viable in a "real world" environment. Finally, we wanted to use this robot to introduce two new function brix that are important additions to our inventory of RoboBRiX.

Figure 1. The assembled Parallax BoeBot



Objective

An important objective, we decided, was to have a quick and easy to build robot that would give rock-solid wall tracking and object avoidance with minimal complexity. While wall tracking would be limited to only one side of the robot (we arbitrarily chose the right side), the robot should be capable of reliably maintaining a steady distance from the wall as it moves forward. It should also be able to avoid the majority of the likely obstructions it might encounter on its travels.

Finally, we wanted to use servos and modulated IR detection devices, since both require specific timing loops. Using these devices would allow us to demonstrate how RoboBRiX make the job of programming so much easier. So, let's get started.

Construction

A year or two ago, our HomeBrew Robotics Club (**www.hbrobotics.org**) made a member bulk purchase of 15 BoeBots from Parallax, Inc. Everyone was so pleased with the BoeBot experience that we thought it would be a good choice for our wall-following platform and, additionally, it would serve to reinforce an awareness that RoboBRiX are platform independent.

The BoeBot has a sturdy, high quality aluminum chassis and comes with two main wheels, a polyethylene ball employed as a tail wheel, and two servos to provide locomotion and differential steering (Figures 1 and 2).

The servos that came with our BoeBot are standard hobby servos with their motion limited to a range of 0-180 degrees and, therefore, we needed to modify them so they could rotate continuously.

The modification procedure is beyond the scope of this article, but the details of this simple hack are available on either the Mekatronix website at **www.mekatronix.com/manuals/misc/servohack.pdf** or the Dallas Personal Robotics Group

website at **www.dprg.org/projects/2003-05a**

Attaching Brix to the BoeBot

We turned, once again, to LEGO® bricks as our favorite method of attaching RoboBRiX to a robot platform. A little bit of two-sided foam tape did a perfectly acceptable job of holding the LEGO brick supports in place on the BoeBot (Figure 3).

To begin, we attached one of the IRProximity2 brix at the front of the BoeBot, oriented forward so that it could detect objects which may appear in the robot's path. Behind this brix, we attached the **Servo4** and connected the two servo motors to it (Figure 4).

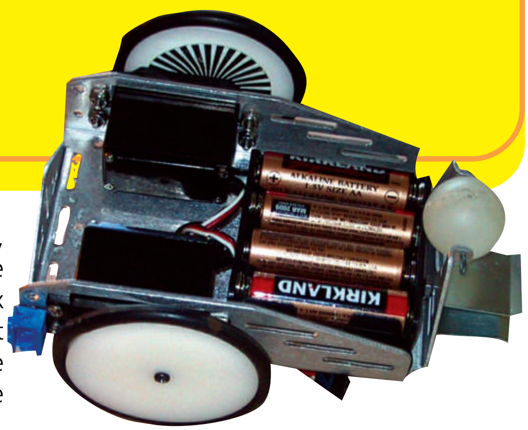
Using more LEGO bricks, we built a simple scaffold to hold the **MicroBrain8** hub and the second **IRProximity2** brix, oriented so that it faces the wall on the right side of the BoeBot (Figure 5). This setup allows for relatively easy physical access to each of the brix used.

The total time it took to build the robot was just about one hour. Of that time, we devoted nearly 45 minutes to assembling the BoeBot itself and 15 minutes to attaching the four RoboBRiX — simple, easy, and quick.

Function Brix Details

Let's begin our discussion of the chosen RoboBRiX with a brief review of how RoboBRiX communicate with each other. RoboBRiX use a simple, asynchronous serial communication method with 8N1 protocol at 2400 baud. Since we decided to use a Parallax, Inc., Stamp BS2 as our microprocessor for the **MicroBrain8** hub, successful two-way serial communication between it and the function brix was simple because we only needed to use two BASIC Stamp commands: SERIN and SEROUT. In practice, therefore, we didn't need to be concerned with any of the brix communication parameters, except the baud rate, since it is one of the arguments used by both of these serial communication commands.

Figure 2. The underside of the BoeBot showing the servo battery pack.



The Servo4

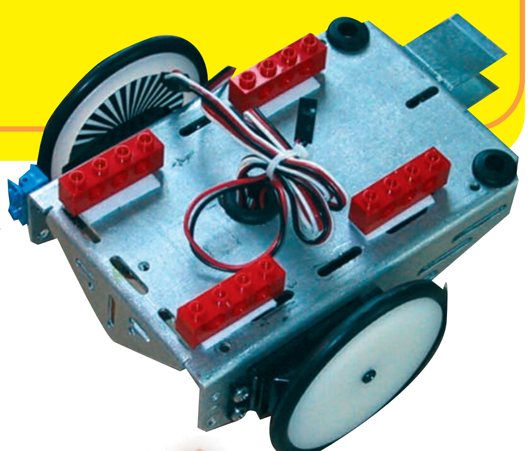
The **Servo4** module can independently control up to a maximum of four hobby grade servos at one time. It makes no difference if it is a servo modified for continuous operation, like those used for robotic locomotion, or an unmodified servo, like those used for position setting.

Servos, as you may recall, will often use a pulse width of 500-2500 µS to set a specific position (or speed/direction, if modified to run free). This pulse must be refreshed at least every 20 µS for the servo to function properly. Ordinarily, this means that the user must fit this critical timing loop into the robot's program so that the servos are refreshed as needed, but without interfering with the remaining code that animates the robot's behaviors.

The good news is that the **Servo4** brix does this critical timing loop automatically and, thus, eliminates any necessity for it to be included in the code that animates the robot's behavior. The animation code is made simpler as a result.

For most hobbyists, the benefit of

Figure 3. LEGO mounting blocks attached with foam tape.



not having to write timing loops would, by itself, make the **Servo4** attractive, but the good news doesn't end there. This brix has a number of other effective and simple to use features as well:

Servo4 Commands: A user has access to all of the **Servo4** features just by sending simple one-byte commands listed in the **Servo4** Programming Table (see excerpt in Table 2).

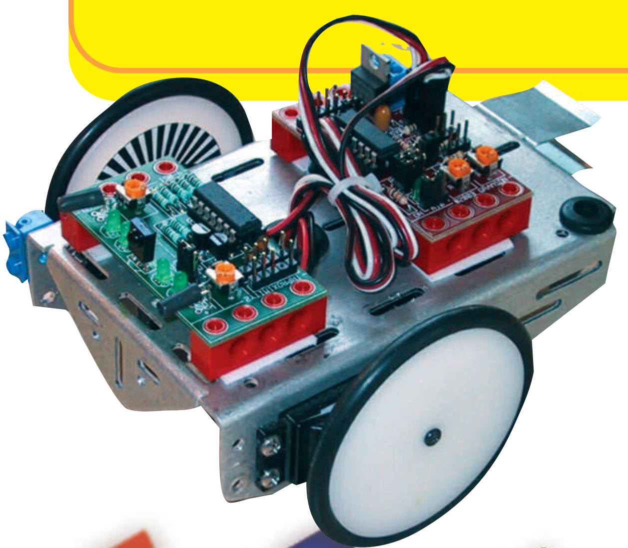
Enable/Disable: As the label implies, servos can be individually or globally enabled or disabled with a one byte command. This is a nice feature that, for example, permits you to set up a servo's initial parameters before turning it on.

Power Supply: Using the logic power supply to run motors often results in the untimely resetting of the microprocessor or other unexpected — and equally disastrous — events. To avoid this kind of interference, the **Servo4** uses an external power source for its operation. The external, unregulated power source is fed to an onboard voltage regulator that, in turn, provides a regulated +5 volts for servo operation.

Calibration: When using a servo for locomotion, it is often difficult to find the precise command value that will bring the servo to a complete halt, without any creeping. The **Servo4** includes calibration circuitry for ports SV0 and SV1, so that the user can calibrate each servo's zero speed point with a simple, one-step procedure.

Stall Monitoring: An important motor property that should not be ignored is the stall current. As a motor slows under increasing load, its system current rises and, theoretically, becomes infinite when the motor completely stalls. By inserting a shorting block on the designated header, the user can begin monitoring the individual current levels of each servo. By adding code to the top-level software written by the user, he or she can create an appropriate response whenever the current levels get too high.

Figure 4. The Forward IRProximity2 and Servo4 brix are installed.



The IRProximity2

This brix was designed specifically to provide a simple, but reliable, method of achieving object detection using an infrared (IR) reflection as evidence of the presence of an object. The detecting system consists of two IR LEDs (940 nm wavelength), modulated at a frequency of 38 kHz to reduce interference from ambient IR light, and a single IR detector, tuned to the same frequency to record any reflections. The two IR LEDs are set about two inches apart on one edge of the brix' longest dimension and the IR detector is positioned equidistant between them.

This brix has a band of detection that ranges from approximately 8 cm to 25 cm. The maximum range of detection can be adjusted anywhere between these two limits. The adjustment is not done, as you might expect, by changing the sensitivity of the IR detector, but is done instead by changing the brightness (energy emitted) of each of the two IR LEDs. Each IR LED has a dedicated potentiometer for this purpose.

The **IRProximity2's** minimum range of detection is primarily a consequence of the distance that separates the LEDs and the detector. When an object is closer than eight cm, the IR reflects at such an acute angle that it misses the detector altogether. Changing the IR LED orientation toward or away from the detector's centerline of detection can reduce or increase this minimum range.

Since this brix uses only one detector, the two LEDs must be pulsed on and off out of phase with one another in order to obtain independent distance readings for each IR LED. Because the two IR LEDs are separated by only about two inches, the two distances recorded for equal intensity settings will not vary greatly, unless the brix is oriented at a fairly large angle of incidence to the object it is "seeing".

Four on-board, visual, green LEDs provide direct evidence about what the **IRProximity2** is seeing and are a useful tool when tuning the IR LED intensities for optimum operation.

When setting the top IR LED to detect far away and the bottom one to detect close in, the visual LEDs can provide a binary readout that is proportional to the distance of the object detected.

If power consumption is critical, the visual LEDs can be turned off by removing the shorting block on the labeled header, but, if power consumption is not critical, the flashing LEDs are a real attention getter and make the robot a lot more fun to watch as it moves about.

Implementation

Servos, modified for locomotion, typically require large changes in the pulse width to achieve noticeable changes in motor speed. Because of this rather poor response characteristic, we decided to limit our motor speed choices to only three states: halt, slow-forward, and fast-forward.

We used the Set High command, found in the **Servo4** Programming Table, to send speed values to the servos. This command takes the binary form `00hh hhss`, where the h's are the binary speed value and the s's designate the servo port.

The value then sent to the designated servo is `hhhh hhhh`. For example, sending the command `0011 1101` ($3d) tells the **Servo4** brix to set the left servo speed to fast forward. It gets translated into a binary speed byte of `1111 1111` ($ff) and designates this speed value for the servo connected to port 1 (SV1).

Remember that, when the robot is moving in a forward direction, one servo is turning clockwise while the other, which is rotated 90 degrees from the first, is turning counterclockwise. Consequently, the right servo's speed values for each speed state are different from the left servo's values for corresponding speed states.

We connected our BoeBot servos to Servo4 ports SV0 and SV1 so that we could use the calibration circuitry to find the precise halt values for each of them. Each servo will have its own unique value, so, if we ever replace one of the servos, we will have to do the calibration procedure again for the replacement.

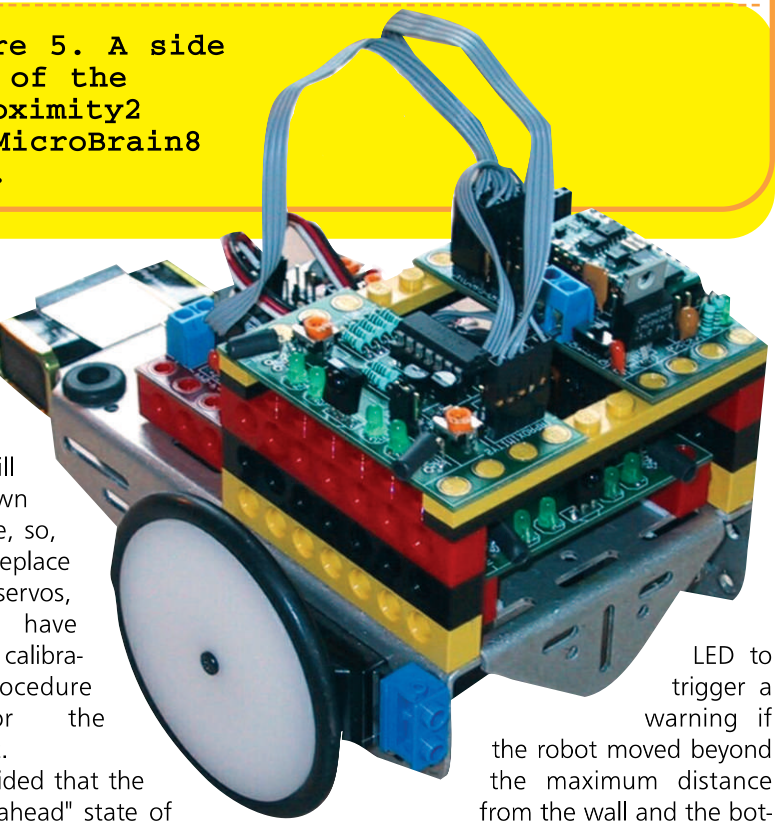


Figure 5. A side view of the IRProximity2 and MicroBrain8 brix.

We decided that the "clear path ahead" state of the front-looking **IRProximity2** should occur only when the IR detector returned a value equal to zero. To get our robot to "round" an object in its path with a good margin of safety, we found that canting the two IR LEDs about 20 degrees to the right of center helped insure it would remain well clear, regardless of the object's geometry. For the side-looking **IRProximity2**, on the other hand, we got the best wall-following results when we toed the two IR LEDs about 20 degrees toward the IR detector's centerline. Table 1 details the robot positions we considered, their associated detector values, and the servo command responses that resulted in the smoothest and most reliable tracking results. Using these values, we were able to create a lane of travel, bounded by maximum and minimum distances from the wall. We set the top LED to trigger a warning if the robot moved beyond the maximum distance from the wall and the bottom one to warn if the robot got closer to the wall than the minimum distance. We arbitrarily connected the forward looking **IRProximity2** to Socket 2 of the **MicroBrain8** and the side looking **IRProximity2** to Socket 3.

Analyzing the MicroBrain8 Code

Having chosen the Parallax BASIC Stamp BS2 for our **MicroBrain8** hub microprocessor, we wrote the top level robot animation code in PBASIC 2.5, using the Parallax Stamp Editor/ Development System, Ver. 2.0 Beta 2.1.

Program Flow

The "top-down" programming order begins with the customary declaration of program constants and

Table 1. Sensor code and behavior chart.

Robot - Wall Orientation	Top or Bottom Sensor Confirmation	Turn Type & Direction	Servo Left Right
Sharply angled in	TopSensor >= $0b	Hard away	LH RF
Slightly angled in	BottomSensor <= 8	Gentle away	LS RF
Sharply angled out	TopSensor <= 8	Hard toward	LF RH
Slightly angled out	BottomSensor >= $0b	Gentle toward	LF RS
Parallel to wall	If none of the above	Straight ahead	LF RF

R = right; **L** = left; **F** = fast; **S** = slow; **H** = halt

variables and the initialization of variables and setting of start conditions (Lines 11-60). The Main Program Loop (Lines 70-120) contains the code which animates the robot's behavior, which we will discuss in more detail below. Finally, the last bit of program code contains the two subprocedures — Sensor Control and Motor Control (Lines 123-132) — that, respectively, gather the IR sensor data and issue speed instructions to the servos. In the Initialization section, you should note that we used the HIGH command to set the **MicroBrain8's** output lines to each of the three function bricks. This is a necessary pre-condition to insure the start of reliable serial communication between the Hub and the function brix.

Main Program Loop

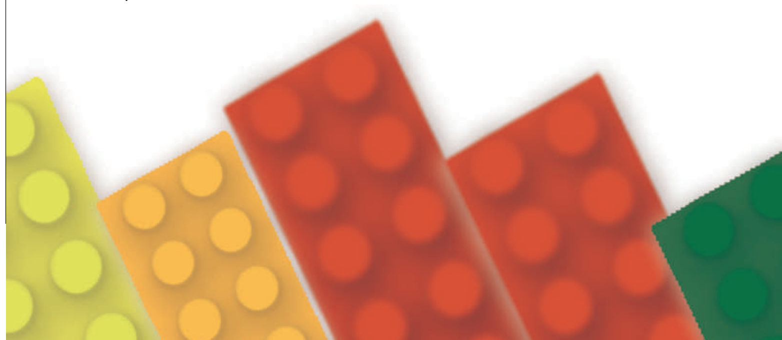
Animation of the robot takes place in the program code beginning with the label "Main Program Loop". The DO - LOOP (Lines 70-120) establishes an infinite loop that continuously manages data gathering from the IR sensors, determines any action to take, and then issues an appropriate speed command to the servos. Since object avoidance has a higher priority than wall-following, the program first interrogates the front sensors for the presence of any objects in the robot's path (Lines 72-82), then issues appropriate servo commands to avoid any that are detected. This part of the program will continue to loop back on itself as long as the front IR sensors report the presence of any path obstruction. Only when the path is clear of obstructions will program execution drop down to the wall-following behavior module (Lines 84- 119).

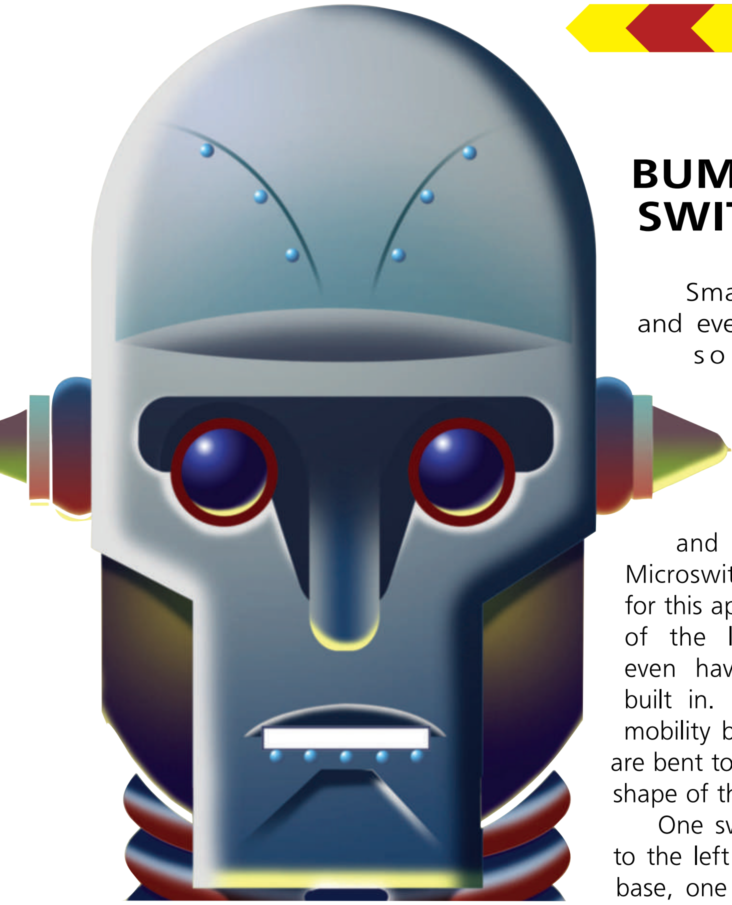
This section of the code uses IF-THEN statements to sift through the wall-reflected IR data received in order to assign appropriate servo motor speeds, in accordance with the table shown in the Implementation subsection above. This program code is quite simple and follows very closely to how we, ourselves, might act if we were controlling the robot's behavior directly. Now, consider how much more complicated this code would be if you had to manage the robot's response behaviors, refresh the servo's pulse rates every 20 µS, and simultaneously measure the width of the reflected IR pulse to determine wall distance.

Conclusion

We were very pleased with how steadily the robot tracked the wall, absent of the "S" turning that was quite apparent in our line-following robot, but even more surprising was how well the robot moved around objects in its path and consistently regained wall tracking once clear. It also was able to detect some objects that we were certain it wouldn't "see", such as a pile of clothes and an occasional table leg.

On the other hand, black shoes were its nemesis. This wasn't really too much of a surprise though — most dark colored objects generally absorb more IR light than they reflect and, therefore, are difficult to detect. In our next RoboBRiX article, we will turn our attention to the **MicroBrain11** RoboBRiX hub that uses the PIC 16F876A as its microprocessor. **SV**





BUMPER SWITCHES

Small robots — and even some more sophisticated types — use switches near the ground to detect objects and avoid them. Microswitches are best for this application. Some of the larger varieties even have long feelers built in. Secured to the mobility base, the feelers are bent to conform to the shape of the robot.

One switch is secured to the left of the mobility base, one is to the right, and, in some sophisticated designs, one is in the rear. Despite the fact that this system is very basic, even advanced robots use it as a backup for light or sonic distance detecting systems.

The switching system only works if an interfacing circuit is provided because, as soon as the switch is released, the direction will turn back and the robot will be caught in an unending back and forth motion.

The circuit diagram (Figure 1) shows the microswitch triggering a 555 timer IC. When the switch is triggered, the IC holds the motor direction relay on for about two seconds. This gives the backup time to pull away, even though the switch has already reset. Resistor R3 adjusts the backup time to synchronize the action with other system parameters. Capacitor C1 may need to be replaced by a smaller value if the operating voltage is more than nine volts.

The relay should be selected to match the system operating voltage. If a rear switch is used, it should set both motors to forward motion by canceling the forward time set by a potentiometer.

STABILITY SWITCHES

One very important element that all robots should be cognizant of is ground continuity. I learned this lesson when my sentry robot took a tumble down a flight of stairs after someone left a stairwell wide open.

If I had installed a ground continuity switch (seen in Figure 2), I would have avoided spending many hours at the repair bench. The best ground continuity mechanism for maneuvering turns is a rolling ball device, seen in the illustration.

Tall robots are vulnerable to being toppled by tables or other obstacles. To guard against this, four mercury switches mounted to a circuit board at angles will detect an alteration in the upright position. Make sure that the angle of the mercury switches is steep enough to

ROBOT Recognition

by Terence Thomas

To a robot builder, nothing is more satisfying than seeing their creation perform well by responding appropriately to its real world surroundings. To achieve this, a number of sensors are required to give a robot cognizance. Even the simplest of robots must be able to move about the world with little or no difficulty.

avoid triggering them at every little bump.

For advanced designs, the mercury switches can provide a trigger to devices that correct the displacement of the robot. For less sophisticated designs, an alarm can be activated to let you know your robot is in trouble.

ROBOT EYES

Another way to keep your robot from bumping into everything is to give it eyes. The two primary components used in electronic eyes are phototransistors and cadmium sulphide photocells. Both are sensitive to light; however, the transistor acts faster, which may not be a factor in your design. Cadmium cells are less expensive, so if you want to set up an array of more than one, you won't need a bank loan.

In the robot eye circuit, we see a cadmium cell connected to a 555 timer (Figure 3). As a robot surveys the territory in front of it, all sorts of light conditions and reflections are encountered. As the robot approaches a wall, the angles of direct and reflected light are cut off until less and less light is detected and a backup pulse is generated. Therefore, robot eyes are best described as dark detectors.

A small tube around the detection device limits the effect of ambient light on the circuit. In the schematic, sensitivity to light is controlled by a trim-pot and should be adjusted to achieve the desired results on your robot. A second potentiometer determines the pulse width of the 555 timer and should be set to provide the proper step signal for your robot. A series resistor serves to limit the output current.

Different amounts of light may strike the surface of the cadmium cell with no effect. When a preset degree of darkness is detected, the 555 timer generates a step pulse. The duration of this pulse is adjustable to accommodate a wide variety of computer sensing situations.

ROBOT EARS

Sound can also be used to trigger a response from your robot. In the circuit illustrated (Figure 4), a condenser microphone mounted on the board can be seen. The 10K series resistor will make the circuit sensitive to only loud sounds, so if you need something more sensitive, increase the value or add a 100K series potentiometer to adjust sensitivity. The 1K output resistor must be included to limit the current from the timer.

Circuitry for the ear is simple enough — a basic "stop what you are doing" on-off response is all that you can achieve. This is justification enough for including it in your robot, especially if it is a

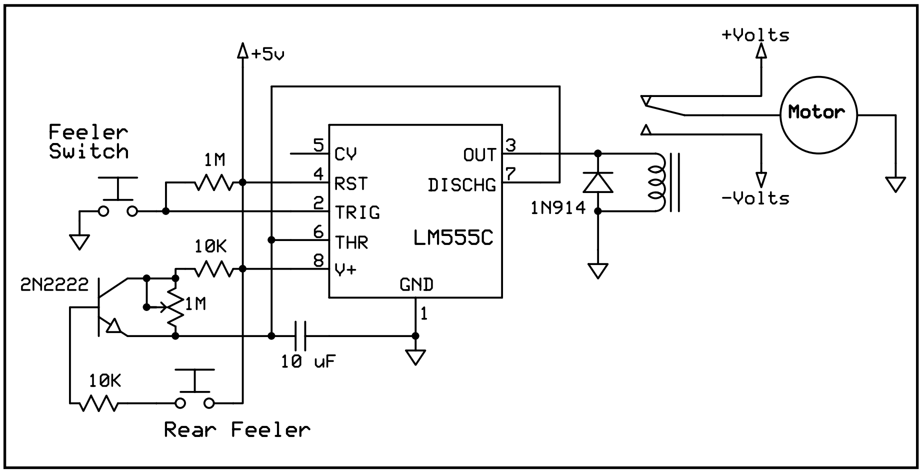
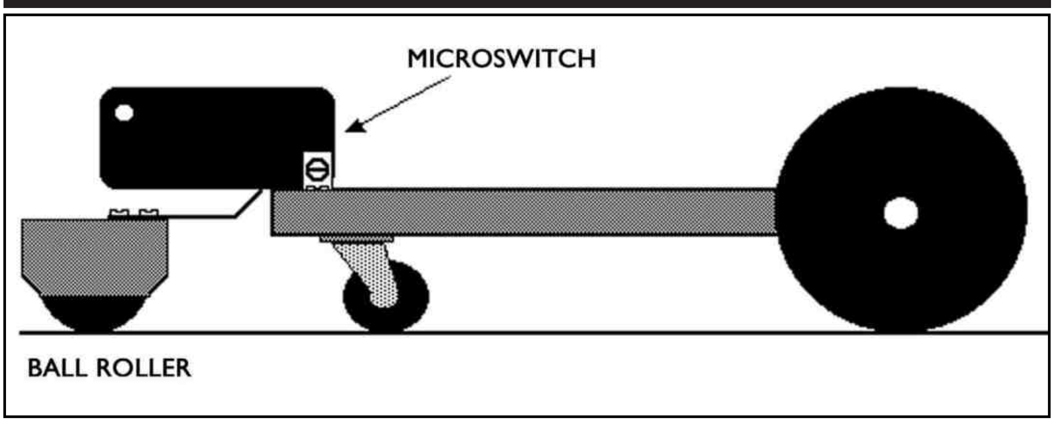


Figure 1. Schematic of a microswitch-based detection control system.

Figure 2. Here is a simple way to implement a ground continuity sensor.



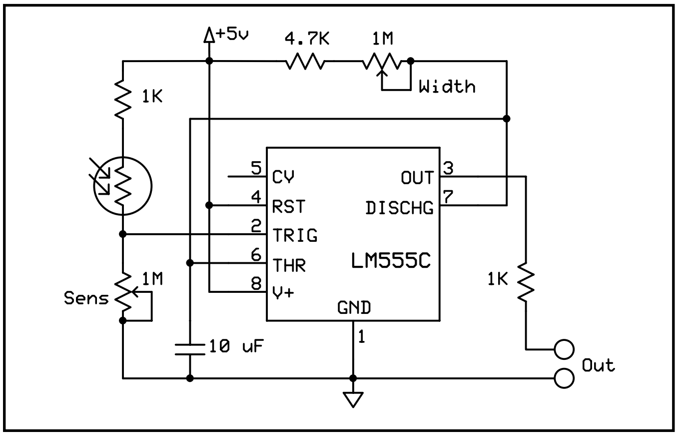


Figure 3. Schematic for a robot "eye".

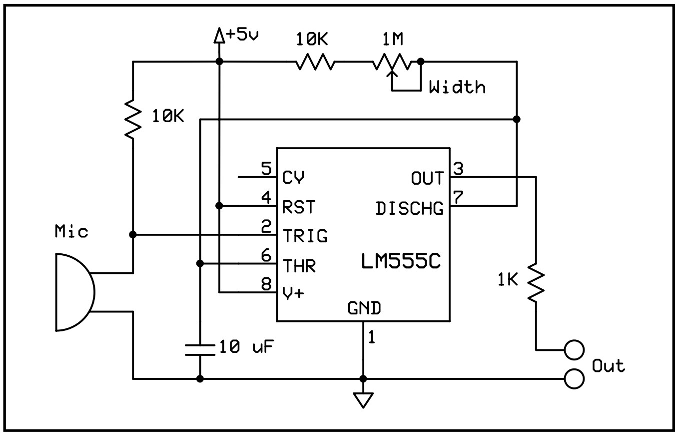


Figure 4. Schematic for a robot "ear".

sentry listening for the sounds of burglars.

With the use of band pass filters, a degree of sound discrimination can be achieved. A flow chart is provided (Figure 5) to show how this is accomplished. Each range will trigger a different function, so this type of robot should be operated in a controlled environment with a specific sound agenda to prevent your robot from acting like a paranoid schizophrenic.

With voice recognition circuits, more advanced responses can be achieved. Even ultrasonic programming has been achieved by using the sonic range finders from old Polaroid cameras.

555

Most circuits in this article use the 555 timer due to the incredible versatility of this device. It can be triggered by just about any device available and only requires a brief needle pulse to engage it.

Output from the timer can be equal to the input pulse or can be set to produce a step pulse, which is adjustable to any length of time necessary for a robot to perform a function. In addition, the timer will operate with a supply voltage range from as low as 3 volts to as much as 18 volts.

The output of the 555 is full power, so a current limiting resistor is absolutely necessary. For interfacing robotic components, it can be said that the 555 timer has no match.

HAND CONTROL

Not all cognizance circuits involve robot mobility. Head, hand, and arm movement require precise control. Sophisticated robotic hands may use conductive vinyl to sense the pressure needed to secure a delicate sample.

Conductive vinyl looks and feels just like the vinyl you find on furniture; the only difference is that conductive vinyl conducts electricity. It acts like a resistor with a value of 330 ohms per square — not per square inch, but per square, no matter what size the square. Increasing the length of a piece of vinyl will increase the resistance to one more suitable for a voltage divider.

OPERATION

A triangular piece of conductive vinyl serves as a voltage divider and is placed near a sensor plate. When a sample is grasped, the pressure displaces

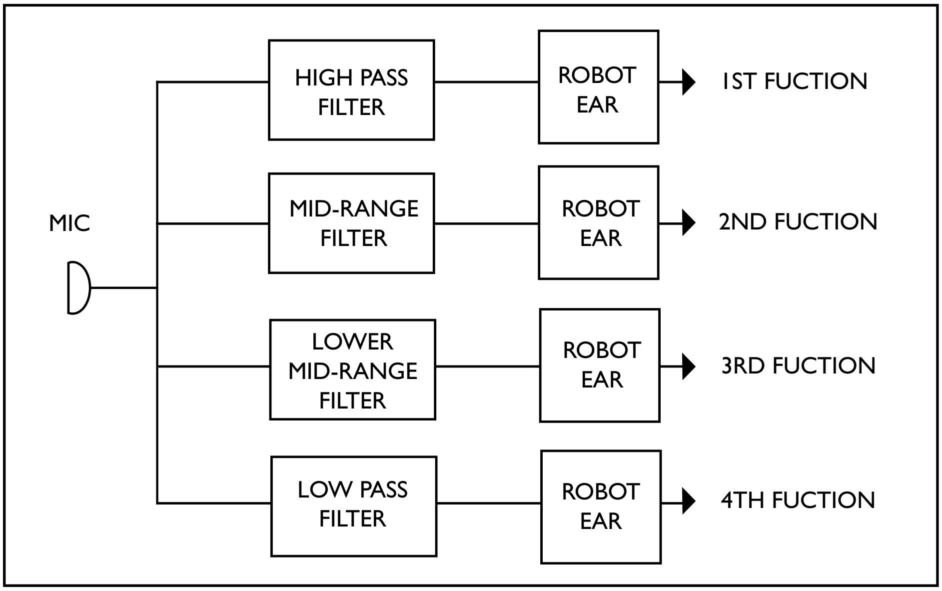


Figure 5. This is the transfer function for a more complex robot "ear".

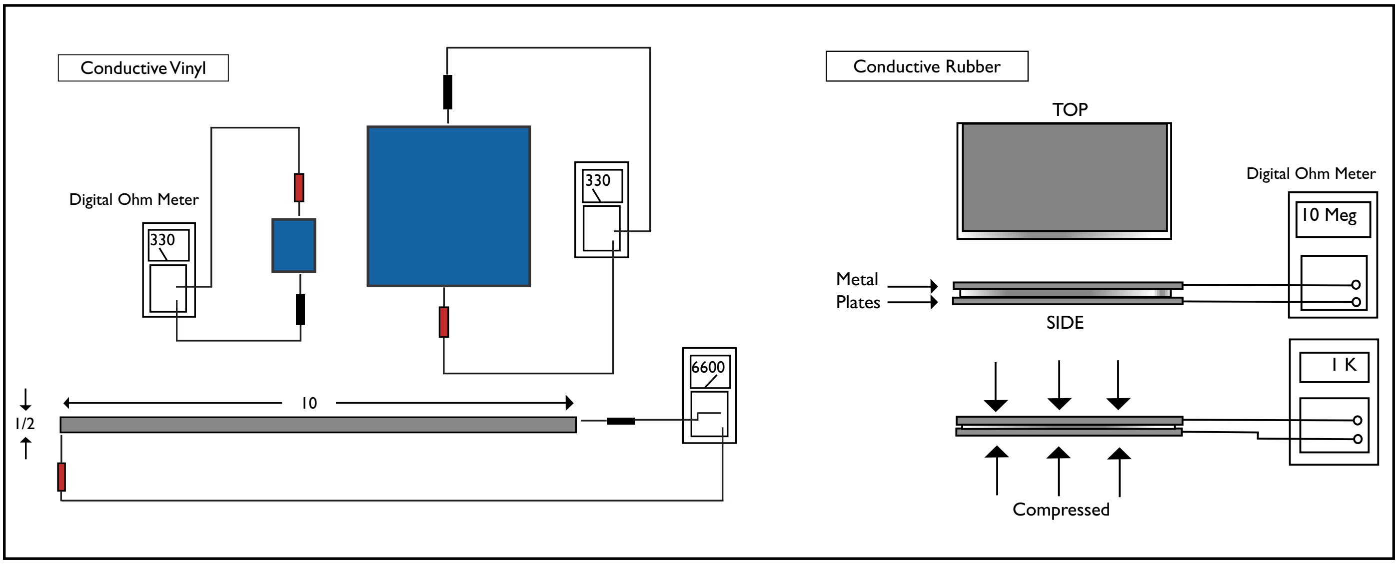


Figure 6. The resistance of these two materials changes either by dimensional ratio or through pressure.

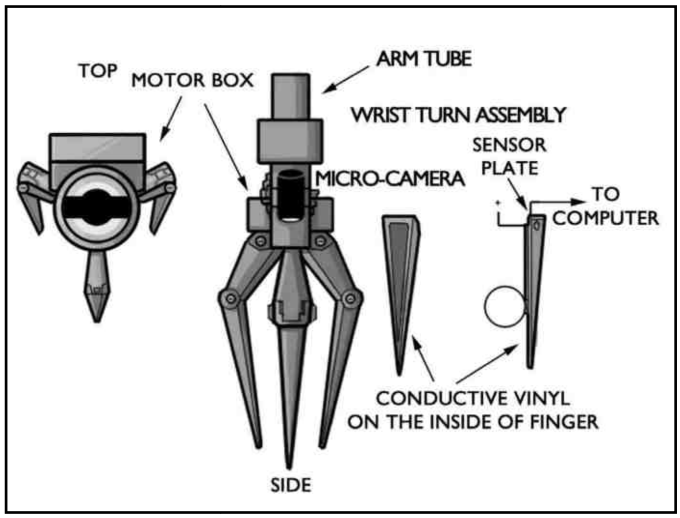


Figure 7. Design of an advanced grasping hand.

the vinyl strip, making contact with the sensor plate.

The resulting voltage level is sampled and, as long as it stays the same, the grip of the hand will stay constant. If the voltage decreases, a signal is sent to the motor box and the fingers' grip is tightened until the voltage stops changing.

As shown in the hand illustration, the motor box, using small gears, can open and close the lower fingers and widen the grip by spreading the upper fingers. The wrist turn assembly can rotate the fingers 356 degrees. Sharp, pointed fingers enable the hand to dig into the ground for buried samples. A micro-camera and light source, mounted in the arm tube, give the controller a perfect view of the collected sample.

Conductive rubber is another substance making an appearance as a potential sensor element for use in robotics. It is made by adding conductive particles to the liquid rubber in the first stages of processing. When the rubber is compressed, the conducting particles become compacted, and a more conductive state is achieved.

It comes in a wide variety of densities that are suited to pressure activating situations from as low as 30 pounds to thousands of pounds — the latter making it very popular at truck weigh stations.

Walking robots can use conductive rubber to indicate contact with the ground, enabling them to more accurately maneuver over rough terrain.

PROGRAMMING

Large industrial robots, like those used in welding cars, use elaborate programming to control every movement, but they also need devices to enable them to evaluate their work and avoid accidents.

CONCLUSION

As with most circuits, careful attention to details will assure a properly functioning device. Experimentation may lead you to different component values specifically suited to your particular needs.

No matter what kind of robot you build, recognition will be a major factor in how well it functions.

Additional foils, diagrams, and circuits for robot sensors can be downloaded from the *SERVO Magazine* website at **www.servomagazine.com** SV

EVENTS CALENDAR

Send updates, new listings, corrections, complaints, and suggestions to: **steve@ncc.com** or FAX **972-404-0269**

March brings us the much anticipated DARPA Grand Challenge, which has become quite controversial due to a series of PR blunders by DARPA that excluded many teams planning to enter the contest; further complications ensued when the secret course was leaked through the web.

Meanwhile, ROBOlympics has run into legal problems with the US Olympic Committee, who object to the character string "Olympics" within the competition's name. They've taken similar legal action in the past against the Special Olympics, the Deaf Olympics, the World Senior Olympics, and other groups (but, oddly, not the Rat Olympics).

*For last minute updates and changes, you can always find the most recent version of the complete Robot Competition FAQ at **http://robots.net/rcfaq.html***

— R. Steven Rainwater

March 2004

9-10 Emerging Robotics Technologies and Applications Conference
Cambridge, MA
This conference explores how emerging robotics technologies are being used to develop new markets and product categories, open additional lines of business, enhance existing product lines, and increase business productivity. (Sponsored, in part, by *SERVO Magazine*.)
www.roboticsevents.com

12-13 AMD Jerry Sanders Creative Design Contest
University of Illinois at Urbana-Champaign, IL
This year, robots will play *Tetris* by forming a completed puzzle out of nine standard *Tetris* shapes on a 36' x 36' playing field.
http://dc.cen.uiuc.edu

13 DARPA Grand Challenge
Los Angeles, CA
This autonomous LA to Vegas cross-country, off-road race has a one million dollar prize. Despite continued setbacks and bad PR, the Grand Challenge is still a go.
www.darpa.mil/grandchallenge

13-14 Manitoba Robot Games
Manitoba Museum of Man and Nature, Winnipeg, Manitoba, Canada
This interesting assortment of robot events includes mini-sumo, Japanese sumo, a robot tractor pull, atomic hockey, and robo-critters.
www.scmb.mb.ca/mrg.html

20-21 ROBOlympics
Ft. Mason Herbst Pavilion, San Francisco, CA
FIRA, BEAM, Mindstorms, and robot combat are all in one place, not to mention the main event — robot builders vs. the US Olympic Committee's lawyers!
www.robolympics.net

28 University of Florida Student Robotic Competition
UF Conference Center, Gainesville, FL
This is the only robot contest you'll see where the robots are required to obey Asimov's three laws as part of the rules!
http://plaza.ufl.edu/niezreck/Robots_Competition_2004.html

April 2004

3 DPRG Roborama
The Science Place, Dallas, TX
Autonomous robots compete in all the usual events: quicktrip, t-time, and can-can.
www.dprg.org/competitions

15-17 FIRST Robotics National Competition
Georgia Dome, Atlanta, GA
Open to the public, this is the championship event for the FIRST student robot competitors.
www.usfirst.org

16 Carnegie Mellon Mobot Races
Wean Hall, CMU, Pittsburgh, PA
In the 10th annual occurrence of the now famous CMU Mobot race, autonomous robots race to complete a complex course, passing through 18" finish gates along the way.
www.cs.cmu.edu/~mobot

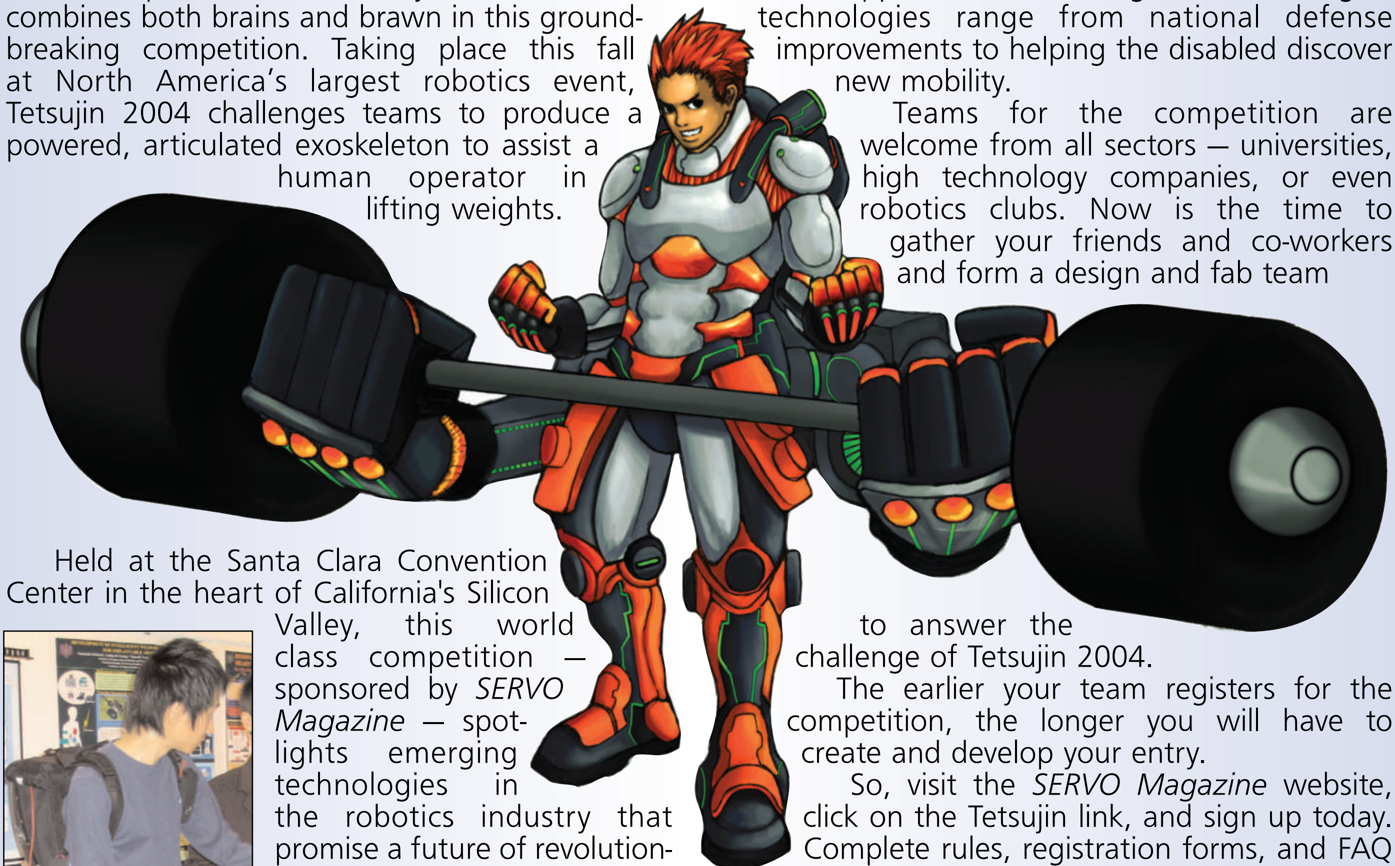
SERVO Magazine Challenges You To

TETSUJIN 2004

October 21-23 — Santa Clara, CA

Man joins machine in a test of physical strength and mental agility ...

Tetsujin 2004 Powered Exoskeletal Competition



The Japanese call it Tetsujin — Iron Man. It combines both brains and brawn in this ground-breaking competition. Taking place this fall at North America's largest robotics event, Tetsujin 2004 challenges teams to produce a powered, articulated exoskeleton to assist a human operator in lifting weights.

Held at the Santa Clara Convention Center in the heart of California's Silicon Valley, this world class competition — sponsored by *SERVO Magazine* — spotlights emerging technologies in the robotics industry that promise a future of revolutionary products, destined to touch all of our lives.

The applications for "augmented strength" technologies range from national defense improvements to helping the disabled discover new mobility.

Teams for the competition are welcome from all sectors — universities, high technology companies, or even robotics clubs. Now is the time to gather your friends and co-workers and form a design and fab team to answer the challenge of Tetsujin 2004.

The earlier your team registers for the competition, the longer you will have to create and develop your entry.

So, visit the *SERVO Magazine* website, click on the Tetsujin link, and sign up today. Complete rules, registration forms, and FAQ are available online.

www.servomagazine.com

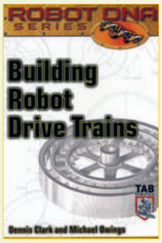
The HAL (Hybrid Assistive Leg) powered unit, developed by the Cybernics Laboratory at the Institute of Systems and Engineering Mechanics, University of Tsukuba, Japan. Learn more about this exoskeletal powered unit online: **http://sanlab.kz.tsukuba.ac.jp/HAL/indexE.html**

The *SERVO* Bookstore

Mind Candy For Today's Roboticist

Building Robot Drive Trains

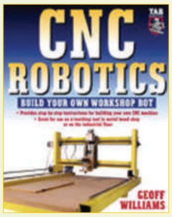
by Dennis Clark / Michael Owings



This essential title in McGraw-Hill's *Robot DNA Series* is just what robotics hobbyists need to build an effective drive train using inexpensive, off-the-shelf parts. Leaving heavy-duty "tech speak" behind, the authors focus on the actual concepts and applications necessary to build — and understand — these critical force-conveying systems. **$24.95**

CNC Robotics

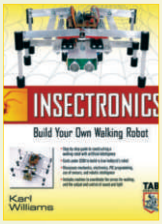
by Geoff Williams



Now, for the first time, you can get complete directions for building a CNC workshop bot for a total cost of around $1,500.00! *CNC Robotics* gives you step-by-step, illustrated directions for designing, constructing, and testing a fully functional CNC robot that saves you 80 percent of the price of an off-the-shelf bot and can be customized to suit your purposes exactly, because you designed it. **$34.95**

Insectronics

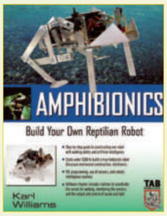
by Karl Williams



This complete project book delivers all the step-by-step plans you need to construct your own six-legged, insect-like robot that walks and actually responds to its environment. Using inexpensive, off-the-shelf parts, hobbyists can "build a better bug" and, at the same time, have loads of fun honing their knowledge of mechanical construction, programming, microcontroller use, and artificial intelligence. **$19.95**

Amphibionics

by Karl Williams



This work provides the hobbyist with the detailed mechanical, electronic, and PIC microcontroller knowledge needed to build and program snake, frog, turtle, and alligator robots. It focuses on the construction of each robot in detail and then explores the world of slithering, jumping, swimming, and walking robots, in addition to the artificial intelligence needed with these platforms. **$19.95**

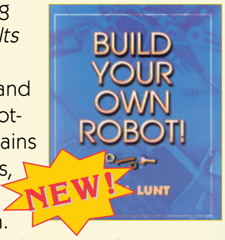
Personal Robotics: Real Robots to Construct, Program, and Explore the World

by Richard Raucci

Personal Robotics gives an overview of available robot products, ranging from simple to complex. Interested readers will be able to find the robot kit that matches their skill level and pocketbook. Other criteria a reader will be able to review include motion systems (from robot arms to robots that roll on wheels or walk on legs), available sensors (from none to a wide range), and programming complexity (how the robot is programmed). If it's not really a robot, it's not in this book. **$25.00**

Build Your Own Robot

by Karl Lunt



This book — a compilation of articles from Karl Lunt's long-running column for *Nuts & Volts Magazine* — is a must-read for all beginner- and intermediate-level robotics enthusiasts. It contains entertaining anecdotes, as well as practical advice and instruction. Possible projects range from transforming a TV remote control into a robot controller to building a robot from a drink cooler. You'll want to build them all! **$34.00**

Robot Builder's Sourcebook

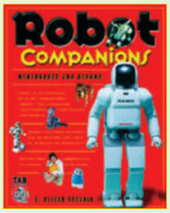
by Gordon McComb

Fascinated by the world of robotics, but don't know how to tap into the incredible amount of information available on the subject? Clueless as to locating specific information on robotics? Want the names, addresses, phone numbers, and websites of companies that can supply the exact part, plan, kit, building material, programming language, operating system, computer system, or publication you've been searching for? Turn to the *Robot Builder's Sourcebook* — a unique clearinghouse of information that will open 2,500+ new doors and spark almost as many new ideas. **$24.95**

We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

Robot Companions

by E. Oliver Severin



With *Robot Companions*, you'll learn how to build your own robot for purposes such as companionship, supervision of the elderly, tutoring the young, doing household chores, and much more. The book delves into essential enabling technologies — such as mobility, voice, communications, touch, sight, and smell response — so you'll understand the mechanics behind form, function, and personality. **$24.95**

Robot Builder's Bonanza

by Gordon McComb

Robot Builder's Bonanza is a major revision of the bestselling bible of amateur robot building — packed with the latest in servo motor technology, microcontrolled robots, remote control, Lego Mindstorms Kits, and other commercial kits. It gives electronics hobbyists fully illustrated plans for 11 complete robots, as well as all-new coverage of Robotix-based robots, Lego Technic-based robots, Functionoids with Lego Mindstorms, and location and motorized systems with servo motors. **$24.95**

SUMO BOT

by Myke Predko / Ben Wirz

Here's a fun and affordable way for hobbyists to take their robot-building skills to the next level and be part of the hottest new craze in amateur robotics: Sumo competition.

Great for ages 14+, the kit comes complete with:

- Pre-assembled PCB
- Multi-function, dual-channel remote control
- Robot hardware, including collision-sensing infrared LED and receivers
- CD-ROM with programming instructions and file chapters of robot building tips and tricks
- A built-in Parallax BASIC Stamp 2 and prototyping area, allowing hobbyists to create their own robot applications without having to purchase additional Parallax products **$99.95**

Robothon 2003

by Karl Lunt

Robothon 2003, the latest in the Seattle Robotics Society's (SRS) international robotics competitions, took place October 24-26, 2003, at the Seattle Center. Attracting over 100 competitors and thousands of visitors, it showcased some of the best in the amateur robotics hobby.

The event offered several different kinds of competitions — from mini-sumo to ant-weight combat. It also gave the public a chance to see the wide variety of robots and robot builders that make this hobby so special. People with no robotics experience got hands-on contact with the Seattle Police Department's bomb retrieval robot and several of the FIRST (For Inspiration and Recognition of Science and Technology) machines, designed and built by local high school students.

The FIRST competition is an excellent national program designed to introduce high school students to a possible career in high tech fields. Those interested in more information, mentoring a FIRST team, or starting a FIRST team in their high school should stop by the FIRST website at: **www.usfirst.org**. One set of awards is presented by a panel of judges, who stroll through the venue, each voting on his/her favorite robot in each of several different categories. As one of these judges, I spent hours checking out the many different designs and talking to the builders.

The vendors' booths at Robothon provide hobbyists a chance to look through the latest products, including motors, wheels, controller boards, and complete robot kits. These tables were favorites for the kids who want to get started in robotics and for the parents who are looking for a way to help their children begin in the hobby.

In the Beginning

Robothon 2003 began on a Friday night with the customary hack session. About 30 people brought robots, tools, computers, and ideas to a workroom reserved by the SRS at the Seattle Center. The group included hobbyists from Canada, Mexico, Australia, and all over the United States.

One of the first robots I noticed was a two-wheeled machine named nBot, built by David Anderson of the Dallas Personal Robotics Group (DPRG). David has a couple of these two-wheeled machines and he brought video of the smaller unit roaming around in his front yard. When I say "two-wheeled," I mean that the robot only touched the ground with two wheels — no skids, no casters, just the two drive wheels. The autonomous robot remained upright by adjusting the wheels' drive based on internal tilt sensors and rate gyros. David is working hard to get the cost of this

Figure 1

Got bombs? The Seattle Police Department sent a few officers and their bomb retrieval robot to Robothon.

Figure 2

A trio of FIRST robots: These machines were built by high school students for the FIRST competition. Several of the students were on hand to drive the robots around and answer questions.

type of balancing sensor system to a low level. nBot's current sensor package costs several hundred dollars — out of the reach of most hobbyists.

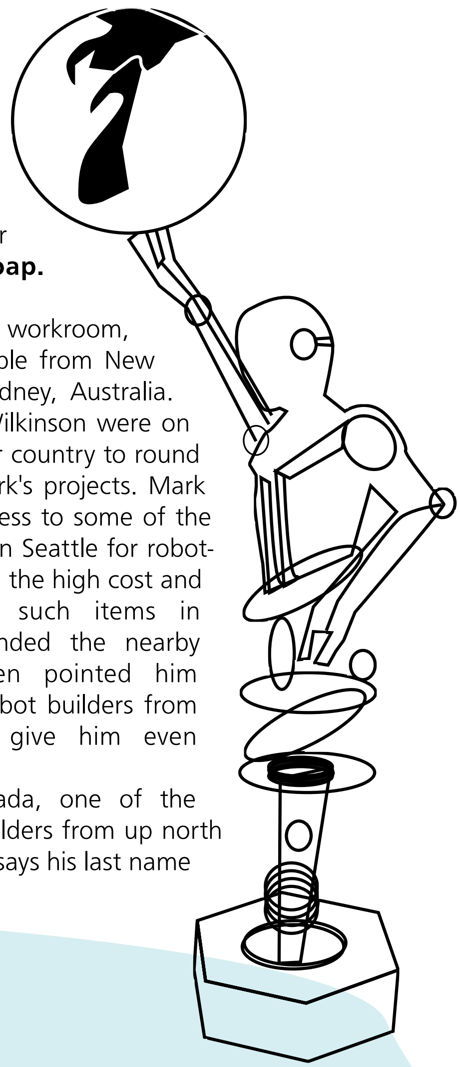
The video of David's robot demonstrated very robust motion over difficult terrain. For example, one sequence showed the robot running down a 30-degree grassy slope at nearly top speed, then hitting a flat stretch of driveway. The machine smoothly corrected for the change in surfaces, never tipping or falling. As this robot scooted around the carpeted floor of the workroom, it would sometimes run into someone's foot, which had been intentionally placed in its path as a test. The robot would either ride a wheel over the foot and keep trucking or back up, switch angles, and try again. However, I never saw it tip over because of hitting an obstacle. By the end of Robothon, David had garnered one of the Judges' Awards for Most Innovative Robot for his work on nBot.

One of the robots at the hack session went on to receive the highest honor in the event, the Judges' Choice Award. Dafydd Walters showed off his OAP machine; OAP stands for Open Automaton Project, a SourceForge project whose goal is, "to engineer modular software and electronic components from which it is possible to assemble an intelligent, PC-based, mobile robot suitable for home or office environments." (See a picture of the OAP machine in "Menagerie" in the February *SERVO*.)

Dafydd (His name is Welsh, by the way, and is pronounced very much like "David".) has crammed some excellent engineering into his three-tiered, cube-shaped robot. The machine stands about 20" high and runs around on two wheels and casters. The core of the machine is a 1 GHz VIA motherboard running Linux. Hooked into this PC is an electronics package, including 12 sonar sensors, passive infrared (PIR) detectors, two FireWire video cameras for stereo vision, text-to-speech output, keyboard, and other items.

Many of the robot's functions are implemented using an onboard I2C network controlled by the motherboard. For example, the 12 sonar sensors are fired in sequence based on commands issued on this network.

One of the factors that led to our selecting Dafydd's project for the Judges' Choice Award was his willingness to share his design with others — an attitude much in the spirit of the SRS. Dafydd has provided all source code, schematics, and design details of his robot on a project page on SourceForge. To view Dafydd's information, aim your browser at **http://oap.sourceforge.net**



While roaming the workroom, I met a very nice couple from New Zealand by way of Sydney, Australia. Mark and Claudette Wilkinson were on a two-week tour of our country to round up robot parts for Mark's projects. Mark was excited to get access to some of the sources we have here in Seattle for robotics parts; he bemoaned the high cost and limited selection of such items in Australia. I recommended the nearby Fry's Electronics, then pointed him towards a group of robot builders from Canada who could give him even more tips.

Speaking of Canada, one of the many visiting robot builders from up north was Dave Hrynkiw (he says his last name

Figure 3

Building the ant-weight arena. Members of the WARS group are putting together the Lexan-enclosed venue for the one-pound combat robots. This was a very popular competition.

is pronounced "Smith"), president of Solarbotics. His company, based in Calgary, has been instrumental in making the BEAM robots accessible to the public. This class of robots is usually solar-powered, using clever analog circuitry to provide behaviors normally found only on machines sporting microcontrollers. Dave was gearing up to run a class the next day where attendees could build their own mini-sumo robots. It was great to see Dave, as I've known of him for years — first as a Forth developer and later as the source of some clever BEAM robot kits.

One of Dave's newest products is a small gearhead system built for the ubiquitous Mabuchi motors. These little motors are dirt-cheap in the surplus houses, but are not usually suitable for robotics, since they spin so bloody fast. Dave has created a set of right-angle gearheads that accept a Mabuchi, clamping the motor into the gearhead with a nylon strap. The gearheads are available in several reduction ranges and the output shaft mates with wheels and couplers from Solarbotics. I especially liked the price — less than $12.00 a pair, including motors. Dave gave me a couple of samples to play with; I'll report back when I get a chance. To check out the company's product line, aim your browser at **www.solarbotics.com**

While you're on that page, check out the details on the Solarbotics Sumovore robot kit. David took one of the two awards for Best Engineered Robot for his design efforts on the Sumovore. It's a nice piece of work, largely because you can change the robot's "brain" by plugging in different microcontroller modules. As designed, it provides an analog-based set of behaviors, but you can also use a BASIC Stamp II module. Solarbotics is promising that two new modules will be available in 2004.

Incidentally, the other award for Best Engineered Robot went to Terry Harmer for his machine, Top Spin. Terry's machine was a three-level platform run by a Hitachi Tiny-8 microcontroller, loaded with sensors and a nice gearhead drive system. (Yes, we handed out two Best Engineered Awards; we are not hobbled by the English language.)

Opening Day

On Saturday morning, the Seattle Center was opened to the public and the Robothon event started in earnest. At the entrance, the Seattle Robotics Society table dispensed information on the events, sold spiffy Robothon 2003 T-shirts and raffle tickets, and answered all kinds of robotics questions. Visitors to this free event could stop by the vendors' tables or check out many of the robots on display from all over the country. Robot builders who wanted to show off their machines could get a table dedicated to their robot in advance, so they could set up informational displays or show the robots in action.

In fact, the winner of the Judges' Award for Most Useful Robot went to Greg Fredericksen for his Freddy the Robot series of projects. Greg has designed the basic Freddy so the robot can perform several different functions, such as talking to kids, detecting a fire and sounding an alarm, or operating a wasp-killer spray can. I was especially impressed with the accuracy of the fire sensor built into Freddy. The sensor, known as the Hamamatsu UVTron detector, can spot open flames as small as a lit match from up to 15 feet away, even in sunlight. The unit is available from Acroname (**www.acroname.com**) for $65.00 each. Greg gave repeated demonstrations of Freddy; the fire alarm and sensor worked perfectly every time.

David Anderson wasn't the only builder with a balancing robot on display. Bob Allen and Ted Larson showed off their balancing, two-wheeled robot run by a PIC microcontroller (also featured in February’s “Menagerie”). This robot also sported a Hamamatsu fire detector from Acroname. The tilt sensor used in the robot came from Parallax (**www.parallax.com)**, also look for the Memsic 2125 dual-axis accelerometer. This $29.00 device measures dynamic acceleration (vibration) and static acceleration (gravity) to 1 mG resolution. Details for connecting the sensor are available on the Parallax site.

Figure 4

Just another day at Robothon 2003. The large glass box on the left is the ant-weight combat arena. The center of the photo shows some of the vendor tables and personal robot displays. Out of the photo to the right is a demo area.

The crowd got a kick out of seeing the balancing robots wandering around the floor. Even those visitors not terribly robot-savvy could see that these machines with only two tires touching the ground were something different. The robots would sway slightly as they sped up or slowed down, giving a graphic demonstration of startup acceleration and braking. David, in particular, spent a lot of time answering questions and demonstrating the strengths and limits of his current design. One of my favorite images from Robothon is nBot standing next to a toddler, himself no taller than nBot, while the child's proud father snapped a digital picture of the two.

In the stage area, the crew from Western Allied Robotics Society (WARS) was bolting together the ant-weight robot combat arena. The arena consists of a six-foot square floor, surrounded by three-foot-tall walls of 1/8" Lexan. The ant-weight combat robots that compete inside this arena may weigh only a maximum of one pound, but they are equipped with some serious weaponry and there's just no sense in showering the audience with hot or sharp robot innards. These are R/C-style machines, each controlled by the team's driver using a standard hobby radio. The contest pits driver skill, design strengths, and construction in a high-speed combat event; each round drew a large crowd.

Additionally, the WARS group set up a special event for the crowd. For a $5.00 fee, anyone in the public could take over the controls of an ant-weight robot for three minutes and try their hand at robot combat. This was very popular and the WARS group kindly donated all proceeds from this rental event to the SRS. If you would like more info on the WARS group, check out their website at: **www.westernalliedrobotics.com**

Day Two at Robothon

Sunday saw several more events, including one I was especially interested in watching — the line maze. As in previous years, the maze was laid out on an eight-foot square white surface with matte black lines, 1/4 inch wide. All turns are right angles and all lines are laid out on a six-foot square grid. This year, the event used a level-three maze; that is, the maze contained both intersections and loops. Had the top robots been evenly matched, the contest might well have been decided by the best algorithm for solving the maze. However, this contest was not even close. Greg Verge — an application software engineer for Cypress Micro Systems here in the Seattle area — took first place with his robot, Cyclops — a stepper-based design built with one of Cypress' PSoC devices. The PSoC (Programmable System on a Chip) microcontrollers seem like a good choice for small robots, as they pack some sophisticated I/O into a small package. The I/O elements can be reconfigured on the fly using your firmware, so one device can provide many different capabilities. For more details, visit the Cypress Micro Systems website at: **www.cypressmicro.com** (For more information on PSoCs, read Al Williams' article in the October 2003 issue of *Nuts & Volts*.) Greg's design, which ran the maze in one-sixth the time of his closest competitor, earned a Judges' Award for Coolest Robot.

Speaking of Coolest Robot, we gave another Coolest Robot Award to Scott Ferguson for a terrific little weapon he built into his small combat robot, Hexy Micro. It isn't easy to build a kinetic weapon into a small machine, as there isn't much mass to work with. Scott solved the problem by installing a pair of powerful coiled springs into his machine. One end of each spring was folded flat inside the robot's body, in line with a slot in the top surface of the machine. These spring ends were held in place with nichrome wire, which kept them pulled down below the surface. When Scott managed to get his

robot wedged under the frame of an opponent's robot, he pressed a button on his control box, heating the nichrome on one of the springs to the breaking point. The spring would snap up out of the robot's frame, hopefully flinging at least part of the opposing robot into the air. Having two springs available gave Scott two chances at taking out his opponent. The two awards for Most Unusual Robot went to deserving machines.

Monty Goodson's robot, Mouse Trap, brought a new weapon into the mini-sumo arena. The size restrictions on mini-sumo (about 3" by 3", but unrestricted height) would seem to rule out the kind of wedge-shaped frame popular in the larger sumo events, but Monty had built a frame with a hinged upper section that folded into legal size. When the robot started a match, it would back up suddenly, causing the upper section of the frame to fall forward at the hinge.

When the upper section hit the contest floor, it created a large and formidable wedge on the front of Mouse Trap. The other Most Unusual Robot Award went to the GYRE project from the University of Washington's Autonomous Robotics and Control Systems (ARCS) Laboratory. GYRE is, "an autonomous, free-flying robot capable of orienting itself using visual cues and navigating in a microgravity environment." And where, you might ask, does one get a, "microgravity environment?" Simple; just hitch a ride with NASA on the Vomit Comet — a research aircraft modified for use as a zero-G testing platform. The GYRE design team was on hand to answer questions, show off the robot, and run some video of the machine in action. You can find more information on the GYRE project at: **http://depts.washington.edu/gyre/**

Also on Sunday, I got an up-close look at another machine that went on to win a Most Innovative Robot Award. Casey Holmes had created a "nano-sumo" robot less than one cubic inch in volume. This tiny machine was equipped with two line sensors to detect the edge of the contest platform, an IR object detector, and a powerful set of motors. Several robot builders had brought nano-size machines to Robothon, and there was an ongoing demonstration of two nano-Sumo machines duking it out on a miniature ring. There is a strong group of nano-Sumo enthusiasts building these tiny machines, and much of what they have done can be applied to larger robots. You can get more information on Casey's machines at: **www.geocities.com/bethmiller805@sbcglobal.net/**

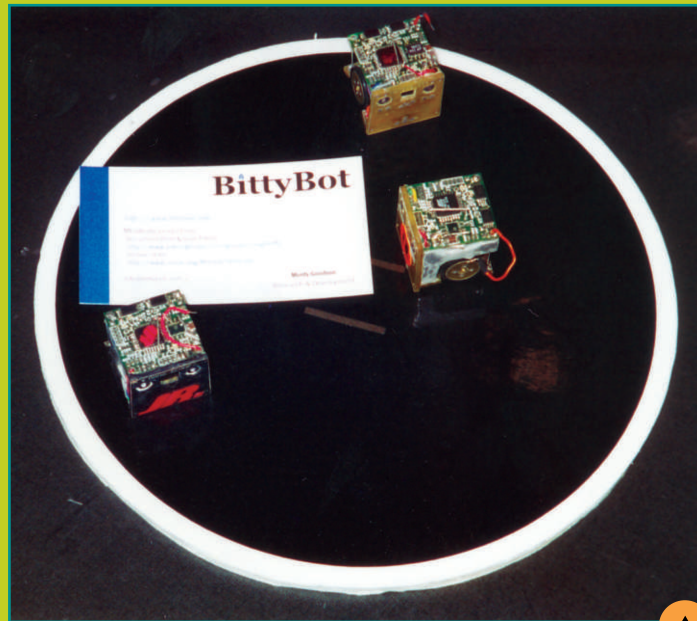


Figure 5

Three robots on a business card — some of the one-cubic-inch nano-sumo robots. One of them, built by Casey Holmes, took a Most Innovative Robot Award from the judges.

Conclusion

I have focused on the Judges' Awards because they were what I spent most of my time doing at Robothon — reviewing and judging robots. However, they don't begin to cover the variety of machines and technologies present. There were simply too many robots running around to mention. Likewise, I didn't have a chance to see some of the contests or to attend any of the robotics papers presented during the event.

My biggest regret is that I didn't spend time with the team from Mexico. This group fielded several entries in the line maze contest and one of the team members, Omar Rodriguez Pureco, took third place, missing second by less than six seconds. I hope that these international 'bot builders enjoyed their visit to Seattle and to Robothon and I hope they will be back for the next Robothon.

The SRS has set up a nice web page about the 2003 Robothon. Go to the main page at: **www.seattlerobotics.org** and select Robothon for a look at the events, the competitors, the robots, and the builders.

There is already talk of the next Robothon. Since the SRS is also not hobbled by strict schedules. I'm not sure when it will be held.

Hosting these events is a lot of work, especially for the main organizers, and it takes some time to recover from the burnout. Pete and Kristina Miles did a heroic job pulling all of this together with a lot of help from many SRS volunteers. The whole group made Robothon 2003 a wonderful event, and I'm looking forward to the next Robothon. Who knows, I might enter a robot next time. **SV**

This Is Where It Starts!

by Bill Woolley

I am the most fortunate guy. Many weekends will find me working in the garage with my two 16-year-old sons. How cool! Of course, I had always hoped for this and, sure enough, it's happening. Having been a gearhead and grease monkey since 1972, when I was 15 and got my first Mustang, it just seemed natural that, when my kids reached driving age, we would work on cars together.

So, here we are, heads together, studying something on the workbench. Well, the Holley on the '65 is acting up a bit, so is that it? Upon closer inspection, the bits spread out on the bench do not resemble a carburetor at all. Actually, they are not even automotive. Well, at least not all of them.

We are together, looking over a couple of big boxes of parts and brainstorming about our next big project — a robotics project! A little wider view shows that we are not alone. My garage is filled with high school students who are all part of Chaparral High School's FIRST Team #1079 (For Inspiration and Recognition of Science and Technology). The six-week build time is here and ideas, sketched on paper, are being loudly debated and transformed into full-size mock-ups for testing. My garage is "Robot Central" because I am the lead engineering mentor for the team, as well as the parent of two team members, and a pack rat.

Whatever happened to the Mustangs? Well, I drive the '65 fastback daily and the '66 GT fastback will wait under a car cover in another part of the garage for a couple more years until the boys go off to school. Of course, I have been told that, by that time, I might be into "real" horsepower, so it may wait a bit longer.

You know what, though? I could not be happier! I feel that those of us who love to make the most out of the brain/hands connection are naturally curious, a bit nuts, and will always be happy working on anything. Cars or robots — it does not matter. Most importantly, I am able to work side-by-side with my sons, all of us immersed in projects together.

Learning is fun, as it should be, since it is a non-stop process. For many years, as far as I was concerned, a speed controller was my right foot. Now, of course, it has a whole different meaning. From the time when my sons were still in middle school and decided they wanted to build a combat robot, our whole family's focus has changed. Many meals are spent discussing robotics ideas and quite a few napkins have been archived, covered with drawings and plans. Their hobby has evolved into not one, but two combat robots. They founded a robotics club at their high school and now, for the second year, have participated in the FIRST program, which is the largest robotics competition in the world. It is all-consuming and we would not want it any other way. We are all having a blast!

I think that all of us who read *SERVO* are really just big kids, in the sense that we gladly accept the great challenges that many people who are too set in their thought processes (if the word thought can be attributed to the limited brain activity of those people with no vision) are incapable of imagining. We know that many of the great discoveries have most assuredly not yet been made. We are the people asking, "What if?"

To be a part of the "outside the box" thinking which creative problem-solving demands is a joy. Being a mentor on a FIRST team involves witnessing and aiding in the incredible growth that occurs in the team members during the six-week build period; it is truly inspiring. That is the word: inspiration. It is the "I" in FIRST and the most important aspect of this wonderful program. Kids are inspired to find, develop, and use their problem-solving skills. Inspiration causes their brains to redline in order to meet the challenges, but they always do it. It makes the light turn on and never go off because they have found a reason to learn. It is awesome!

I am truly a most fortunate guy. I turned my hobby of working on cars into a wonderful, satisfying career in the racing business. I love what I do, so it really is not work. Plus, it is exciting to see this happening for my sons, as well — even if they couldn't care less about cars and racing. They are into robotics and are preparing to go to college to further their dreams. Whether it is through building LEGO Mindstorms robots for PAReX and the Science Olympiad, tweaking their 30 and 60 lb. combat robots for the next BotBash, or working together with the team on their FIRST robot, it is all learning and fun, ensuring a great and rewarding future. In addition to all of that, I get to be an active part of their endeavors. You just cannot beat that!

So, will I stop my involvement with robotics in a couple of years when my sons go off to college? Hardly. I am hooked and having way too much fun. I also would not leave the other students on the FIRST team, to whom I feel an obligation; I have already been

asked this question by some of the younger team members. Our team is small, but growing, and our impact on our school, local businesses, and community is ever-expanding. The privilege of helping kids enlarge their horizons through the realization that everyone has the ability to think and create is such a gift. Seeing the team members through their senior years with the team is a must. Hmmm, this could go on a while ...

Besides, I have this idea that has been rattling around in my head for a solar-powered, autonomous creature that would wander around my yard. That might really freak out the horses, though. **SV**

Author Bio

Bill Woolley has been in the automotive racing business for 25 years; he has worked with Cosworth Racing, Inc., for the past 15 years. "Robot Central" is located in Temecula, CA, where he lives with his wife, Debbie, and twin sons, Bryce and Evan.

Advertiser Index